Design and development of digital image processing techniques for latent fingerprint enhancement and separation

Aditya Singh

2008

Aditya, S. (2008). Design and development of digital image processing techniques for latent fingerprint enhancement and separation. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/41428>

<https://doi.org/10.32657/10356/41428>

Design And Development Of Digital Image Processing Techniques For Latent Fingerprint Enhancement And Separation

Aditya Singh

School Of Mechanical And Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Master of Engineering

2008

Abstract

This project deals with image enhancement of fingerprints with a focus on latent prints. Latent prints are involuntary prints and hence are often of bad quality with large regions of non-recoverable information. While a standard inked and rolled print may contain essential data points, like the minutiae, in the region of forty and above, for matching of latent prints as little as eight points of match are required.

To be used in an automated software based matching system, it is essential to provide prints of certain minimum quality to ensure that the match results obtained are limited to the most likely candidates. Image enhancement is done in order to ensure this minimum quality. Fingerprints can be taken as patterns of ridges and valleys and any method that improves the clarity of this pattern can serve as the image enhancement stage.

One of the key elements for the process of enhancement is the orientation map. Orientation map refers to the directional information of the print ridges. In case of latent fingerprints, presence of degradations like strongly oriented noise in the background, random noise elements or presence of a second print makes the directional data available for print region more ambiguous. Without any operator intervention, determining the directional information for these regions becomes extremely difficult. Discounting the effect of random noise, we are still left with regions of multiple true orientation values. To allow the process of enhancement and to minimize operator intervention, it is essential that this ambiguity is minimized for the print image and in the case of overlapped prints, even be utilized to find details pertaining to both the constituent prints.

This work tries to study the process of orientation map detection in a systematic way with focus on latent prints and oriented texture patterns in the background. The project starts with a review of literature detailing the present methods utilized for the purpose of fingerprint identification and classification, the software implementation for automated systems and processing techniques for image manipulations that make this automation possible. Various steps required in this automation process have been listed spanning from image entry to extraction of relevant information, design of filters, feature extraction and finally matching and verification.

The next part of the work deals with development of standard algorithms using a script based interface which in our case is MATLAB. A sample database was created with relevant test cases dealing with the kind of distortions mentioned earlier as our primary area of study. Results of standard algorithms on these test cases were then analyzed for future course of development. The third part dealt with the development of a software framework incorporating all the standard algorithms for generation of orientation maps. Care was taken to make this as flexible as possible and preference was given to free and open source libraries with possible future work in mind. This part has been described in chapter 4 of this report.

Based on the analysis of the results on our test cases, more efficient and more specific algorithms were then developed in the last and final part. These new algorithms serve as a hybrid between different processing techniques for generation of orientation maps and retain the relative advantages of them. Emphasis was on accuracy of detection and not on speed or resource requirement as these methods were defined for specific and targeted cases that do not form the bulk of the operation in the field of fingerprints.

Acknowledgments

First, being a part-time study towards a Master's degree, help and patience of my supervisor Assoc. Prof. Seah Leong Keey and my co-supervisor Assoc. Prof. Murukeshan along with their constant encouragement has been a fundamental factor in completion of this research. I'll like to thank them for their time and also for adjusting their schedules to meet me in order to keep a tab on the progress of the work.

Thanks are also due to Mr. Dinish and Dr. Chao, our team members working in the field of fingerprint during my time at the university. Discussions with them have been very useful and working with them was a pleasure.

I will also like to thank the technicians in charge of my lab for their availability whenever I had problems with any equipment.

Contents

Abstract	I
Acknowledgements	III
Contents	IV
List of Figures	IX
1 Introduction	1
1.1 Introduction	1
1.2 Fingerprint Evidence	3
1.3 AFRS Systems: Existing Methodologies	6
1.4 Objective	9
1.5 Scope	10
1.6 Organization	11
2 Fingerprinting - An Introduction	13
2.1 Introduction	13
2.2 Fingerprint Matching	15
2.2.1 Ridges and Valleys	15

CONTENTS

2.2.2	Matching Criteria	16
2.2.2.1	Fingerprint Classification	17
2.2.2.2	Fingerprint Matching by Minutiae Map	19
2.2.2.3	Other Methods of Fingerprint Matching	23
2.2.3	Error Terminology	25
2.3	Automatic Fingerprint Recognition Systems	26
2.3.1	Orientation Detection	27
2.3.2	Filtering	29
2.3.3	Binarization	29
2.3.4	Thinning	30
3	Oriented Texture Image Processing	31
3.1	Introduction	31
3.2	Orientation Detection	33
3.2.1	Gradient Based Orientation Detection	33
3.2.1.1	Edge Detection	33
3.2.2	Filter Based Methods for Orientation Detection	37
3.2.2.1	Rotating Kernel	37
3.2.2.2	Stock and Swonger	39
3.2.3	Other Methods	41
3.2.3.1	Gorman and Nickerson	41
3.2.3.2	Mardia	42
3.2.3.3	Mehetre	42

CONTENTS

3.3	Orientation Modeling	43
3.4	Quality Measure	44
3.5	Filter Design	45
3.5.1	Gabor Filters	45
3.5.2	Bandpass Filter	46
4	Project Components and Software Development	48
4.1	Introduction	48
4.2	Tools	49
4.2.1	Hardware	49
4.2.2	Software	49
4.2.2.1	Visual C++ .NET	50
4.2.2.2	GDI+	51
4.2.2.3	Winforms	51
4.2.2.4	OpenCV	51
4.2.2.5	MATLAB	53
4.2.2.6	Verifinger	53
4.3	Project Work	53
4.3.1	Phase 1: Fingerprint Detection	54
4.3.1.1	Hardware Interfacing	54
4.3.1.2	Automation	55
4.3.1.3	Images	56
4.3.2	Phase 2: Test Algorithm Development	56

CONTENTS

4.3.2.1	Rotating Kernel: A Software Implementation	57
4.3.2.2	Results of standard algorithms	59
4.3.3	Phase 3: BiometricOCV Software Development	63
4.3.3.1	Software Description	64
4.3.3.2	Structure	66
4.3.3.3	Usage: Recipe Settings	69
5	A novel orientation detection method based on quality measure	73
5.1	Introduction	73
5.2	Orientation Detection	75
5.3	Proposed Method	76
5.4	Orientation Filtering	77
5.5	Experimental Results	78
5.6	Conclusion	78
6	A scheme for detecting multiple orientations in an image	81
6.1	Introduction	81
6.2	Second Maxima	82
6.3	Discriminator and Direction Sensitivity	82
6.4	Results	83
6.5	Conclusion	87
7	Conclusion and roadmap for future development	90
7.1	Introduction	90

CONTENTS

7.2	Quality Based Orientation Detection	91
7.3	Second Pass Algorithm	92
7.4	Conclusion	93
	Bibliography	95
	Publications	102
	Appendix	102

List of Figures

1.1	(a) Latent and (b) Enrolled fingerprint from the Mayfield trial	4
1.2	(a) Latent and (b) Inked fingerprint from the McKie trial	4
1.3	ImaQuest Overlapped Case	6
1.4	Imaquest Overlapped Case (FFT)	7
1.5	ImaQuest Enhanced Image	7
1.6	An example pattern (left) and its approximate FFT Representation.	8
1.7	(a) Original print. (b) and (c) selective enhancement results on overlapped region.	9
2.1	Comparison of ridge details present in fingerprints.	15
2.2	Skin cross section of a fingertip	16
2.3	Shows the six basic types for fingerprint classification. (a) Arch, (b) Tented Arch, (c) Right Loop, (d) Left Loop, (e) Whorl and (f) Twin Loop.	17
2.4	Accidental Case	18
2.5	Type Line.	19
2.6	Basic Fingerprint Features	20
2.7	Minutiae (a) Ridge Bifurcation (b) Ridge Ending.	21
2.8	Neighborhood based operation.	22

LIST OF FIGURES

2.9	(a) Fingerprint with 'X' marking the core point (b) Circular Tessellations. [1]	24
2.10	FAR vs. FRR curve.	26
2.11	Overview of the AFRS System	28
3.1	Comparison of different kinds of textures.	32
3.2	Edge	34
3.3	Edge: First Differential	34
3.4	Edge: Second Differential	35
3.5	Rotating Kernel Mask	38
3.6	(a) Ideal edge as a solid line and edge as a slow change shown in dashed line.(b) Edge lines encountered in our case of textures.	39
3.7	Stock and Swonger Mask	39
3.8	Orientation Image Calculation	43
3.9	Gabor filter at 45^0 (a) A Gabor Filter Response. (b) Bandpass characteristics in orthogonal direction with Gaussian pattern in the direction of the filter.	45
4.1	Microsoft Visual C++ .NET Express Edition	50
4.2	Schematic for (a) homodyne and (b) heterodyne assisted phase resolved technique	55
4.3	Sample Images obtained by phase resolved setups	56
4.4	Original Test Image	57
4.5	A representation of a typical input case	58
4.6	Masks at three orientations	58
4.7	Orientation detection process	59

LIST OF FIGURES

4.8	Orientation detection result for 'Stock and Swonger' filter	60
4.9	Orientation detection result for 'Rotating Kernel' filter	61
4.10	Orientation detection result for 'Sobel' mask	62
4.11	Response at a pixel to different masks of 'Rotating Kernel'	62
4.12	Main Window	64
4.13	Function Panel	64
4.14	Recipe Panel	65
4.15	Main Recipe	65
4.16	Analysis Recipe	66
4.17	Analysis Window	67
4.18	Program Structure	67
5.1	Fingerprint Matching	75
5.2	Gradient detection	76
5.3	Process flow of orientation detection methods	77
5.4	Results of the standard and the proposed method	79
5.5	Result of the standard and the proposed method on phase resolved images	80
6.1	Two pass second maxima querying method for coarse orientation map . .	82
6.2	Coarse orientation map output of the three methods	84
6.3	Indicative filter output of the three methods	85
6.4	Results of the quality based and the second pass method on phase resolved images	86

LIST OF FIGURES

6.5	Coarse orientation results of standard, quality based and second pass algorithm on overlapped prints	88
6.6	Indicative filtered results of standard, quality based and second pass algorithm on overlapped prints	89
7.1	Process improvement of quality based orientation detection	92

Chapter 1

Introduction

Summary: This chapter gives an overall picture of the project and the motive behind choosing this field of study. Case studies highlighting the problems faced by law enforcement agencies have been included to lay emphasis on the need for more specialized software tools and image processing algorithms. The field of interest is then narrowed down to the subject matter of this report and a brief description of other tools available for achieving similar objectives is given. The conclusion of this chapter gives an overview of the structure of this report.

1.1 Introduction

Fingerprints are one of the most popular forensic tools used by law enforcement agencies throughout the world to investigate crime scenes. They constitute as an essential piece of evidence in several court cases and are also used in gathering a list of suspects by investigating agencies in case of unsolved incidents. Though several relatively newer forensic methods have been introduced which have given enforcement agencies more tools with which to investigate and gather evidence, fingerprint evidence is still perceived as an infallible proof of guilt by various juries and judges. With a few qualifications, it can also be a sole criteria for establishing the culpability of a defendant. [2, 3]. Recent developments, introductions of DNA fingerprinting [4, 5], fiber analysis [6, 7] and technologies

CHAPTER 1. INTRODUCTION

---

which are capable of providing clues from analysis of trace amounts of elements at the scene of crime have led to a severe dependence on the experts in these fields. Since these new methods are often results of a particular established field of science switching over to forensics (eg. DNA fingerprinting), their acceptability as evidence rests on a strong foundation. Fingerprinting on the other hand, started as a forensic tool [8] and its acceptability has been based primarily on empirical studies. It still lacks any other proof of individuality [9]. This fact has been used time and again to challenge its acceptability as crucial piece of evidence in various trials [10] which have had their basis on Daubert criteria [8]. Though in the review of literature, one has not come across any case in which such challenges have stood in the way but it certainly has created a need for standardizing the method for declaring identification based on the prints.

Till now, this has been based on the opinion of qualified fingerprint analysts who will stand by the uniqueness of the print and have often termed the process of identification used as a perfect science which has zero possibility of error. Reduction in probability of erroneous identification is achieved by using multiple experts to certify the prints as a match.

Challenges to the admissibility of fingerprints in court have gained from the fact that despite of being extremely rare, false identifications have occurred [11, 12]. These cases have been popularized by the media and critics of fingerprint identification and have had a negative impact on the general acceptability of fingerprints.

Police departments have tried to standardize the minimum number of minutiae [13] identified in a latent print to make them admissible as evidence but the very degraded quality and the low number of prints that are available at the scene leads to heavy reliance on human experts to correctly decipher the unclear parts and get best possible numbers of minutiae information from them. The standardization process though, is still not complete and there is a difference in the definition or description of a perfect match across various agencies [8, 14]. Even with involvement of multiple experts, false minutiae have been identified in rare cases much akin to what happens in software based detection algorithms

CHAPTER 1. INTRODUCTION

---

though till now the probability of such a thing happening in software is far higher as the element of intuition is missing and approximations to methods applied by human experts have had success to a limited extent.

Advantages of having software-based methods, which are highly automated and cover most of the steps, are manifold. Software techniques have the potential to improve print quality and have a high repeatability rate, a crucial factor for this field of evidence. Apart from automation, software tools can be and have been an immense help to experts. The massive database searches can be done with a high FAR or false acceptability rate to return a minimized set of enrolled fingerprints as possible matches reducing the manual workload. Enhancement tools for latent images can be used to make the images clearer leading to better visibility and thus more accurate matches and further reduction in error rates. Most software tools also have a quality measure to exclude non-ridge areas and areas of low quality ridge details from a fingerprint image to reduce the possibility of false minutiae detection in automated processes.

The next section will deal with real life cases which have challenged fingerprint cases and in which fingerprints have been a crucial factor but scant in their availability due to various factors.

1.2 Fingerprint Evidence

One of the recent trials involving erroneous identification is the 'Mayfield Trial' [12]. In this trial Brandon Mayfield, an ex US army officer was held in Portland Oregon on charges of being involved in the Madrid train bombing. The crucial piece of evidence was the fingerprint found on a bag that was supposedly used for carrying the detonators for the bombs. Mayfield was later released when the Spanish law enforcement authorities found only eight points of match between the fingerprints while getting a positive match for another man Ouhane Dound. The US FBI had found fifteen pieces of match between

CHAPTER 1. INTRODUCTION

Figure 1.1: (a) Latent and (b) Enrolled fingerprint from the Mayfield trial

Figure 1.2: (a) Latent and (b) Inked fingerprint from the McKie trial

the two prints.

The fingerprint images from the Mayfield trial are given in figure 1.1 [15]. As can be seen from the image of the latent print, when compared to a rolled print, the quality is extremely low. This holds true for most of the latent prints obtained from the scene of crime. The average size of the latent print is supposed to be 21.7 percent of an actual rolled print. Latent prints are also usually deformed. Elastic nature of skin makes these deformations non uniform and hence difficult to model or compensate for. [8, 16, 17].

Another famous case of wrong identification is the Shirley McKie [18] trial where a posi-

CHAPTER 1. INTRODUCTION

---

tive identification of the print was doubted by outside experts while the identifying agency, SCRO or Scottish Criminal Records Office kept defending its decision. Shirley McKie was finally released after the intervention of media that highlighted the case and then of the local politician and external fingerprint experts. The prints from the trial are shown in figure 1.2 [15].

In this trial, the experts judged the two prints shown as coming from the same person. The error as indicated by the arrows is the minutiae point in each print that has no corresponding point in the other print. Getting a minimum number of matching points for two prints is not enough. Sometimes, as in the case above, a singular point of mismatch is enough to dismiss the match.

A third case involving wrong fingerprint matching was trial of Stephen Cowan. After an erroneous identification, Stephen Cowan was sentenced for shooting a police officer and served six years in jail before a subsequent DNA analysis absolved him of his crime. Later, it was also found that the latent print from the crime scene did not match his fingerprint [11, 19].

Overlapped prints also come up in tough cases such as one mentioned in the Internet news magazine "The Weekly Detail" in its issue on February 11 2002. The case involved three prints from a twenty-year-old case with one set being overlapped. Overlapped region was ignored, as sufficient details were available outside the region to eliminate the suspects. This issue does however indicate the presence of the problem of overlapped prints and thus the possibility of them being a hindrance or causing wrong identification similar to cases mentioned in this section.

Overlapped prints have been mentioned often as a problem faced by fingerprint experts and are normally just ignored during investigations, as degradation may be deemed severe enough. In literature too, they have been mentioned as a problem area [20] . There exist certain AFRS systems which even claim to deal with the problem as shown in the next section but even after an extensive search in the scientific literature dealing with the field

CHAPTER 1. INTRODUCTION

Figure 1.3: ImaQuest Overlapped Case

of fingerprints, it's rare to find a work in this field. Being one of the standard types of degradations, there is a need to develop a documented and standardized method to deal with this kind of problem. Also, there is a need for software tools to deal specifically with this issue.

1.3 AFRS Systems: Existing Methodologies

Two systems in existence were found to mention overlapped prints as problems they deal with. The first one is the ImaQuest software that is sold by crimescope [21]. The brochure details FFT, or fast fourier transform, as the tool to deal with the problem. The images are shown in Figure 1.3.

The FFT of the image is shown in Figure 1.4, the blocked regions in this frequency domain representation can be seen as two dark regions close to the center of the image.

Blocking of the FFT is equivalent to suppressing certain frequencies present at a particular orientation in the actual image. The result of this is shown in Figure 1.5.

The results obtained look satisfactory with details for the print in the middle sufficiently visible. Objections, however, can be raised on the methodology due to its hit and trial nature and universal effect of blocking FFT components affecting the whole image in case

CHAPTER 1. INTRODUCTION

Figure 1.4: Imaquest Overlapped Case (FFT)

Figure 1.5: ImaQuest Enhanced Image

CHAPTER 1. INTRODUCTION

Figure 1.6: An example pattern (left) and its approximate FFT Representation.

of non ideal cases unlike the one above.

An FFT representation becomes easier to understand on looking individually at the representations of different frequency components of the image i.e the spatial frequency. Consider the pattern shown in Figure 1.6(a). Its approximate FFT amplitude image is shown in Figure 1.6(b). This is also the frequency component blocked by the ImaQuest software for its sample case.

Elimination of the side lobes visible in the FFT spectrum will lead to the elimination of the oriented lines in the actual image. ImaQuest example, by elimination of these lobes, suppressed the ridges oriented in the corresponding orientation. Parts of the middle fingerprint would also have been eliminated if a similar oriented pattern was present. Region based selective elimination is a possibility that can be considered for reducing this but it will still constitute a tedious process based on guesswork.

The second software that addresses the problem of overlapped prints is Hyundai AFRS system called automatic fingerprint identification system or AFIS in their brochure. The picture of the overlapped region shown as the example is given in figure 1.7. The technique used to enhance the image has not been mentioned but similar results had been obtained by using oriented filters with appropriate direction maps in the initial stages of this research work. These results will be discussed in Chapter 5, Chapter 6 and Chapter 7 dealing with

CHAPTER 1. INTRODUCTION

Figure 1.7: (a) Original print. (b) and (c) selective enhancement results on overlapped region.

preliminary results, final outcome and subsequent conclusions.

1.4 Objective

Overlapped prints are a challenge recognized by literature and commercial software distributors dealing with fingerprint image enhancement and recognition. For a field as crucial as fingerprint recognition that is facing challenges on admissibility and has a zero tolerance for erroneous results, it is essential to develop tools which can help software systems as well as human experts in getting better and more accurate results. Lack of scientific literature on the actual solution or a pathway to the solution is an issue that should be looked into in order to find effective solution for such cases. This research work tries to

CHAPTER 1. INTRODUCTION

---

pay attention to this particular area by means of applying existing methodologies, studying their effects on the problem cases and modifying the methodologies to reach a possible solution that involves minimum intervention by the end user. Thus, the primary objectives of this research are -

- Evaluation of current software algorithms dealing with orientation detection and development of new ones specific to the task of enhancing latent fingerprint images.
- Design and development of software tools to carry out latent print enhancement. These tools will have a capacity to perform analytical tasks using currently accepted methods and also the new methods proposed. Since documentation is a primary concern for any forensic application, these tools will have a capacity to display the intermediate steps leading to the result.

1.5 Scope

In the following chapters, various methods employed for the orientation mapping and filter design have been mentioned. This work tries to employ these methods to obtain orientation maps of prints that overlap with another oriented pattern, including another print, in the background. The work focuses on an in house database collected for this study and an analysis of results, with a view on possible separation of the two prints, is carried out. The main emphasis remains the orientation maps to be generated and a foray into the field of filter design is done purely for the purpose of analysis of the results.

A general breakdown of the tasks to be carried out for this research has been itemized below-

- Perform a literature review of the field of fingerprinting to obtain sufficient information on the tasks of fingerprint identification, classification and their use in the field of forensic studies.

CHAPTER 1. INTRODUCTION

- Perform a literature review of the image processing techniques commonly utilised in analysing oriented textures. Oriented texture being the class of images which cover fingerprints. Also, to do this with the view on automation of fingerprint identification task.
- Develop scripts in MATLAB for the standard algorithms to analyse the results of their application on fingerprints and the effect of different parameters.
- Develop a software for carrying out analysis of fingerprint images, noise elements in the prints and different degradations.
- Based on results of all the previous steps, design new algorithms which are more suitable for the task of gathering orientation data from noisy prints.

1.6 Organization

This chapter has highlighted the need for enhancement tools catering to various degradations in order to make the identification process by human examiners more accurate and if possible eliminate the chances of error. Cases involving difficult prints where identifications were made in error have been highlighted not as a criticism of the experts but as a reason for investing more time in finding methods to eliminate the effect of noise elements present in latent prints. Special emphasis has been put on the case of overlapped prints that forms the area of focus of this work.

Chapter 2 and Chapter 3 deal with an overview of literature in the field of fingerprinting and special tools of image processing employed in this field respectively. An attempt has been made in these chapters to introduce the most basic topics relevant to this field, and the knowledge of which, are essential for becoming familiar with the terminology and processes to be used.

The tools, software and libraries used to carry out this research have been mentioned

CHAPTER 1. INTRODUCTION

---

in Chapter 4. Chapter 4 also introduces the matlab scripts used to develop the standard algorithms and the in house software developed during the course of this research work. Configuration of this software via recipe settings is then explained and will be a useful reference for any future work to be carried out in this area.

Chapter 5 describes a new orientation detection methodology for fingerprints and also presents the results in comparison to standard method. Chapter 6 focuses more on the field of oriented texture (i.e. fingerprints) in the background of another print and suggests a method using the new algorithm described in Chapter 5 to an advantage. It also presents the comparison between standard, Chapter 5 based and the the preliminary results of the modified method. The drawbacks, areas of possible improvement and as such a future pathway for productization are mentioned in the concluding chapter, Chapter 7.

Chapter 2

Fingerprinting - An Introduction

Summary: This chapter deals with the introduction to the science of fingerprints. The process of using fingerprints to establish the identity of a person has a long history and is used by police departments throughout the world. An effort has been made to give an overview of the basic processes involved in the use of this biometric technique for purpose of establishing identity.

2.1 Introduction

Fingerprints are used for the purpose of identity establishment in diverse scenarios. Nowadays it's common to see fingerprint sensors attached to mobile computing devices, security locks and access systems. Fingerprints are also being used as a forensic tool by law enforcement agencies. This use of fingerprints in diverse scenarios, though essentially similar, can be broken down into two main activities, 'Verification' and 'Identification'.

Verification is the process of comparing a single template against a freshly acquired print. This is common in systems that involve a single user or in authentication system where a person already claims an identity and the print acts like a password or a signature. This will be applicable to security needs in mobile computing devices and composite access systems, i.e. access systems with two or more methods of inputs together.

CHAPTER 2. FINGERPRINTING - AN INTRODUCTION

When a single print is compared against a database of fingerprints, the process is called 'Identification'. This is true in case of forensic evidence. Identification, in this scenario, is done using a print obtained involuntarily. These prints can be highly degraded and require a process of enhancement before they can be used. The kinds of degradation and issues that are dealt with when using a print of this kind are listed below [22] -

- Size: The print obtained is partial. A latent print normally has only one fifth the surface area of an inked or a scanned print.
- Location: Information distribution in fingerprints is not uniform and one region may contain more points of interest when compared to another region with the same area. Thus the print obtained might not have enough details.
- Surface: The surface on which the print was deposited might be dirty and obscure several features of the print.
- The print might be smudged or smeared.
- The print might be overlaid or under laid by other prints.
- The print might be deposited on a noisy surface like wood that will contain ridges and grooves, which can be confused with those contributed by the actual print.
- The medium in which the print may be deposited may interfere with its definition. This medium can be blood, oil, sweat and others.
- The amount of pressure and the direction of the pressure exerted when the print was formed dictate the distortion of the print.
- The surface on which the print was formed might be curved or irregular.
- Procedure of lifting the fingerprint will lead to a limited loss of data.

An example of a print obtained voluntarily is given in Figure 2.1(a). The ridge details are clear and making an identification using this print is thus, correspondingly easy. To

CHAPTER 2. FINGERPRINTING - AN INTRODUCTION

Figure 2.1: Comparison of ridge details present in fingerprints.

emphasize the contrast in the kind of problem faced and the level of difficulty involved, figure 2.1(b) shows a print obtained involuntarily.

The major challenges faced are in the field of identification which is also the field this research work is geared towards. Keeping this in mind, the problems mentioned and the underlying assumptions about the requirements of a computer based fingerprinting system for the rest of the chapter, will refer to 'identification' and not necessarily 'verification'.

2.2 Fingerprint Matching

2.2.1 Ridges and Valleys

Fingerprints are made up of ridges and valleys on the surface of the fingertips. The presence and persistence of these structures can be better explained by looking at the cross section of the skin layer. The two main layers present in the epidermis, or the outer layer of the skin, are -

- Stratum corneum
- Stratum mucosum

CHAPTER 2. FINGERPRINTING - AN INTRODUCTION

Figure 2.2: Skin cross section of a fingertip

Stratum corneum is the outermost layer and is uniformly spread. The ridges on the surface are present due to the folds in the underlying stratum mucosum. The various layers of the skin have been shown in Figure 2.2 [23] borrowed and modified from the book “The Science of Fingerprints” published by the Federal Bureau of Investigation, United States Department of Justice (FBI) [23].

This essentially means that a small or minor damage to the skin surface will not erase the pattern or damage it. As soon as the top surface heals, it will regain the ridge and valley structure and will follow the folds in the lower mucosum layer. Only if the injury is serious and deep enough to actually damage this underlying layer will the ridge structures be damaged.

2.2.2 Matching Criteria

Matching criteria for fingerprints is decided by the algorithm used. Several different representations exist to carry out or assist the task of matching and these include frequency domain signature of the two prints [24,25], correlation based matching techniques, count-

CHAPTER 2. FINGERPRINTING - AN INTRODUCTION

Figure 2.3: Shows the six basic types for fingerprint classification. (a) Arch, (b) Tented Arch, (c) Right Loop, (d) Left Loop, (e) Whorl and (f) Twin Loop.

ing the ridges for core-delta pairs [23] and minutiae based matching methods [26–29]. A description of the newly introduced terms, 'Core', 'Delta' and 'Minutiae' is in order here. The next two subsections, section 2.2.2.1 and section 2.2.2.2, have a detailed description in their actual context. Briefly, 'Core-Delta' representation is a global descriptor for a fingerprint. Core-Delta representation also includes another indispensable element called the 'Type Line' [23].

2.2.2.1 Fingerprint Classification

Fingerprints can be classified on the basis of few common patterns that are present. The use of patterns is done basically for classification and has more importance in the field of indexing and database storage [30]. These classifications constitute a global descriptor of the print and the focal points of the patterns are called the core and minutiae. The outermost ridgeline that surrounds the pattern area is called the type line [23].

CHAPTER 2. FINGERPRINTING - AN INTRODUCTION

---

10. Accidental.

Figure 2.4: Accidental Case

Though the number of basic patterns are not fixed or standardized and vary in different literatures [23, 31, 32] with several classifications and sub classifications within, for the purpose of introduction, the following six are mentioned.

- Arch
- Tented Arch
- Right Loop
- Left Loop
- Whorl
- Twin Loop

Fingerprints from each one of these types are shown in figure 2.3 [31]. Apart from these, there are certain cases in which the classification cannot be made, as the pattern does not subscribe to any one single type. These cases are referred to as 'accidentals'. An example of the accidental case is shown in figure 2.4 [23].

The definition of type line given earlier is a simple one, in practice its not this straightforward to describe. In its simpler form, a type line will look something similar to Figure 2.5.

CHAPTER 2. FINGERPRINTING - AN INTRODUCTION

Figure 2.5: Type Line.

The dark outer line in the figure is the type line. It's marked as 'B' in Figure 2.5(b). As can be seen, it completely covers the entire pattern area, marked as 'A' in Figure 2.5(b). The illustration given is for the classification type 'Loop'.

Delta point and Core Points can be grouped together as points of 'Singularity'. Points of singularity are those places in the ridge structure of a fingerprint where the orientation, or the ridge direction, vanishes [33].

A short description of the delta point would be the point at which the diversion of the type line towards the pattern area starts. Similarly, core point can be described as a point on the ridge ending or the innermost loop of the ridge in the pattern area that is farthest from the delta point. Both of these are rudimentary definitions given just to draw a picture of what the points of singularity are. A better definition with working examples for the classification can be obtained from [23].

At this point, a look at Figure 2.6 will clarify the definitions of the more essential features involved in fingerprint matching and classification.

2.2.2.2 Fingerprint Matching by Minutiae Map

Minutiae refer to one of the local features present in a fingerprint image. As mentioned before, a fingerprint image is made up of ridges and valleys with two parallel ridges sepa-

CHAPTER 2. FINGERPRINTING - AN INTRODUCTION

Figure 2.6: Basic Fingerprint Features

rated by a valley. This is true for most of the print region but at certain points, the ridges may bifurcate or abruptly end. These points are termed the minutiae points.

Several minutiae points may be present in the fingerprint image and as single entities; they do not define the print unlike the core and delta points. On a typical fingerprint of good quality, the numbers of minutiae point that can be located are in the region of 40 to a 100 according to FBI(US) book 'The Science of Fingerprints' [34] and 60 to 80 according to reference [35]. For fingerprint matching, the relationship between these points is used [36,37].

Automatic Fingerprint Recognition Systems, which will be defined in a greater detail in the later part of the chapter, map these features and their interrelationships to provide a distortion invariant matching criteria. These distortions can occur from different factors like the orientation of the input fingerprint, presence of a skew towards a direction during the fingerprint obtaining process, presence of smudges and occlusion in the image or false minutiae detection due to the presence of small scars on the tip of the finger [1].

The number of minutiae points required declaring a fingerprint as matching the one in the database can vary depending on what the court allows or what is set as the criteria in an authentication system. In certain cases, ten points of match are taken as sufficient to

CHAPTER 2. FINGERPRINTING - AN INTRODUCTION

Figure 2.7: Minutiae (a) Ridge Bifurcation (b) Ridge Ending.

declare a match between the prints [13]. Cases have also been seen where a print has been declared a match due to sufficient points of minutiae present but have later been found out to be different due to the presence of more minutiae points that did not match in the same print. Overall, minutiae based matching techniques are one of the most popular methods and are very robust [38].

Figure 2.7 shows the two types of minutiae entities present. A ridge bifurcation is branching and parting away of a single ridgeline into two and a ridge ending is sudden termination of a ridge. The placement of these details on the tip of the finger is unique by empirical evidence [9]. Using the ridge following algorithms can do their detection by computational methods.

The steps required from the step of scanning in a fingerprint to getting the minutiae map have been listed below -

- Image Enhancement
- Orientation Field Estimation
- Ridge Detection
- Ridge Thinning
- Minutiae Detection

CHAPTER 2. FINGERPRINTING - AN INTRODUCTION

Figure 2.8: Neighborhood based operation.

All these image-processing routines have been elaborated in the later section dealing with a basic Automatic Fingerprint Recognition System (AFRS).

Once all the possible minutiae have been detected after the process of thinning and neighborhood based approaches, i.e. a three connected neighborhood shows the presence of a bifurcation while a single connected neighborhood indicates a ridge ending, their coordinates are mapped with information on the local ridge orientation as well as the kind of the minutiae [29]. These mappings are later used to provide matching against a map obtained from the print of interest.

Figure 2.8 shows an approximate binarized and thinned representation of the ridge bifurcation and ridge ending shown in figure 2.7(a) and figure 2.8(b) respectively. A neighborhood based operation will be a single pass operation which increments the count for each pixel location based on the number of pixels with binary level one found in an eight connected neighborhood. Neighborhood operations are based on either a four connected neighborhood of the pixel with row and column co-ordinates $[x, y]$ i.e. pixels $[x+1, y]$, $[x-1, y]$, $[x, y+1]$ and $[x, y-1]$ or eight connected neighborhood which consists of pixels $[x+1, y+1]$, $[x-1, y+1]$, $[x+1, y-1]$ and $[x-1, y-1]$ apart from the pixels in the four connected neighborhood. Looking at figure 2.8(a) with the darkened pixel and a shaded re-

CHAPTER 2. FINGERPRINTING - AN INTRODUCTION

---

gion around it, it can be seen that the pixel of interest has three pixels with value one in the neighborhood. This indicates the presence of a ridge bifurcation similarly in figure 2.8 (b), the darkened pixel has only one pixel of value one in its shaded eight pixel neighborhood indicating a presence of ridge ending.

2.2.2.3 Other Methods of Fingerprint Matching

Though minutiae based matching techniques are the most commonly used method for matching fingerprints other methods based on fingerprint properties do exist with their relative strengths and weaknesses. Co-relation scores between two prints are one such method and require a perfect alignment of the prints to give a match score. As is apparent from the description, this method will fair badly if the print used suffers from non-linear deformities when compared to the print enrolled in the database. Also, the co-relation process involves a huge amount of computation considering the two inputs are images of fingerprints at 500 dpi that will be compared against each other with considerations of linear distortions and rotation angles.

A modification of this method is hybrid matchers [1, 39] which will use minutiae information to align the two prints thus cutting down on the time required to complete the alignment before going for the correlation scores. Another method has also been proposed which uses the combination of minutiae information and associated local ridge characteristics to design a matching algorithm based on what the paper terms - 'Global Comprehensive Similarity' [40].

Jain et al [35] proposed a method of fingerprint matching using a quantification termed Fingercodes . The process starts with locating the core point of the fingerprint and centering on this point, dividing the entire print into non-overlapping regions by the use of circular tessellations. This process of assigning circular tessellations centered on the core point is shown in Figure 2.9. Once these overlapped regions have been designated, each region is assigned a value based on the local ridge characteristics for example the 'ridge

CHAPTER 2. FINGERPRINTING - AN INTRODUCTION

Figure 2.9: (a) Fingerprint with 'X' marking the core point (b) Circular Tessellations. [1]

frequency'. The matching is now reduced to obtaining the values of this fingercode and comparing it against the values obtained from an input fingerprint and calculating the Euclidean distance between them.

An improvement over this process was claimed by Arun Ross et al [1] which uses square tessellations to obtain non-overlapping regions and then obtain the ridge feature map similar to the fingercodes. With this method, the necessity to detect the core point of the fingerprint, which by itself is a difficult task, is done away with. Instead, the minutiae features obtained in the print are used to align the enrolled and input image for comparison based on their feature vectors.

Method proposed by Ross et al [1] is designed for situations where the number of minutiae detected in the print is not enough to provide the required level of consistency in fingerprint matching. The specific case mentioned is that of solid state sensors used for obtaining the print where the quality tends to be lower than that obtained using optical reflective methods [41] currently used in government bodies like the Singapore Immigration and Registration.

Both the methods based on tessellations mentioned above make use of filter banks for selectively enhancing ridges of a particular orientation. The number of filters used is eight

CHAPTER 2. FINGERPRINTING - AN INTRODUCTION

---

and consist of evenly spaced, in orientation, Gabor filters and thus the set of feature vectors obtained per print numbers eight also. Gabor filters and their properties will be discussed in more detail in Chapter 3.

Similar to the idea of fingercodes, another global descriptor based on ridge density maps has been proposed in [42]. This relies on the fact that the density, or the distance between two ridges, varies from location to location in a print. Since this feature is deemed essential in literature relating to the modeling of fingerprints and in some dealing with filter design [43], it is argued that by itself, it should be able to provide a unique representation. Even in this method local information, i.e. minutiae, is used for the overall alignment. Another method based on using multiple templates thus forming a distortion-invariant image is proposed in [44] which claims to outperform fingercode based methods. A matching based on Discrete Cosine Transform (DCT) features has been proposed in [45].

Apart from identification, there also exist methods which are more suitable for the core purpose of verification [46,47].

2.2.3 Error Terminology

Software based matching techniques, unlike the institute of professional fingerprint experts who need to be certain about the matching for it to be admissible as evidence, don't claim a perfect match and make use of statistical tools to show the performance of their algorithms.

Two crucial terms associated with this are the 'False Acceptance Rate' or 'FAR' and the 'False Rejection Rate' or 'FRR' [31, 48, 49]. Against a database made up of enrolled fingerprints, if a print is deemed a match when it is not supposed to be, it contributes towards the 'FAR'. Conversely if a print is rejected and is not identified even when the fingerprint is present in the original database, it contributes towards the 'FRR'. The chart in figure 2.10 [32] shows the relationship between 'FAR' and 'FRR'.

As can be seen, there is a trade off involved and decreasing any one of these error rates

CHAPTER 2. FINGERPRINTING - AN INTRODUCTION

Figure 2.10: FAR vs. FRR curve.

will lead to a corresponding increase in the other one. The curve is useful for system evaluation and for tuning. Cases in which security is a must and unauthorized entry has to be brought down to minimum will use a biometric algorithm at a high 'FRR' and a low 'FAR'. In case of police databases however, where a list of all the possible suspects has to be generated, a low 'FRR' and a high 'FAR' system will be more useful. The curve above has the FAR vs. FRR curves for two different Software Development Kits (SDK) that have been developed by the same vendor [49].

This kind of curve results from the fact that any matching algorithm has end result quantified in terms of a score that can be compared against a variable threshold to pass it as a match or a failure.

2.3 Automatic Fingerprint Recognition Systems

With the increase in the number of prints in the database, examining the entire database physically for the purpose of identification, even after incorporating the classification scheme, is an impossible task for the law enforcement bodies. The staggering size of

CHAPTER 2. FINGERPRINTING - AN INTRODUCTION

---

the databases, which in a centralized system like the Federal Bureau of Investigation or the National Identification Documentation in case of Singapore can run into millions of prints, require a computer assisted method. With these quantities as a major consideration, the requirement for the matching algorithms has speed as one of the major factors [50].

Minutiae based methods mentioned in the previous section are fast and have good robustness. AFRS systems have a primary requirement of returning a set of fingerprints that most closely match the one supplied as the input.

An overall view of the process can be gained by looking at figure 2.11. The process starts with a fingerprint input. The next stage can either be an image quality assessment routine or an image pre processing routine for image normalization or contrast improvement. These, apart from the quality assessment, are standard procedures for most of the computer vision applications and hence have not been shown as stages in the figure.

From these stages, the processed image is passed on to the orientation detection stage. Detection of image orientation has been discussed at detail in Chapter 3. It follows a few basic methodologies mostly based either on the output of filter masks directly or by the use of edge detection masks (gradient operators).

The definition of the stages of AFRS system has been provided below -

2.3.1 Orientation Detection

Since fingerprint can be taken as a set of curved line segments, it follows that a description or association of an orientation parameter is possible at each point. This helps us in extending the concept of filtering out noise, an undesirable artifacts in single dimensional signals, to the two dimensional data of a print as the objective now shifts to preserving a sinusoidal signal [34] present orthogonal to this orientation in a fingerprint neighborhood. Orientation information, apart from the help it provides in filter design, is also helpful in establishing the fingerprint classification system in an AFRS system as it can be used as

CHAPTER 2. FINGERPRINTING - AN INTRODUCTION

Figure 2.11: Overview of the AFRS System

CHAPTER 2. FINGERPRINTING - AN INTRODUCTION

---

input data in a 'Poincare' [32, 51] function to determine which group the print belongs to. Also, it is essential in forming models of the print and in filling in the pattern, i.e. guessing the pattern, by use of equations such as Reaction-Diffusion [52] functions which have found use in describing other similar unique patterns seen in nature like the stripes of a zebra. Orientation information, along with minutiae, is also used for matching the prints as mentioned in section 2.2.2.3.

2.3.2 Filtering

Filtering is an essential process to make the image 'usable'. This is because apart from the voluntary prints obtained for database entry in an AFRS system, the other crucial input remains involuntarily left behind 'latent' prints at the crime scenes. These could have suffered severe degradation and may require processing for removal of noise and fixed patterns in the background before they can be used. In fact the very issue of having a quality determining routine in AFRS systems is to achieve the objective of limiting false detections to within acceptable limits.

2.3.3 Binarization

Binarization just refers to reducing the grayscale depth of the print to a single bit or two values '0' and '1', the 'binary' system. This is done to assist in feature extraction which is based on the neighborhood based operation. The main parameter for this process is the threshold value which can be either based on the statistics of the whole image, be a fixed pre determined value or be based on statistics of a region in the image and hence vary from one point to another.

Image statistics refers to the histogram data, average grayscale value of the pixel intensities and such.

CHAPTER 2. FINGERPRINTING - AN INTRODUCTION

2.3.4 Thinning

Thinning is the process of progressively shaving of the width of line segments in the image till every feature is exactly a pixel wide.

Next chapter deals with the design of enhancement filters for the fingerprints with an added focus on getting the input parameter of local orientation. A good orientation map for the print is an effective step in suppressing the background information and is the main area of interest in this research work.

Chapter 3

Oriented Texture Image Processing

Summary: Oriented textures refer to images with a repeatable pattern to which a direction can be assigned in a neighborhood. The chapter covers a basic introduction to various processing tools available for manipulation of this kind of image data. Specific attention is given to orientation detection techniques that give us the direction map of the Images.

3.1 Introduction

A dictionary definition of texture is 'something composed of closely interwoven elements; specifically: a woven cloth' or 'the structure formed by the threads of a fabric'. This deals more with the sense of touch rather than the visual texture which is often dealt with in engineering problems. For a picture, the definition given by Merriam-Webster is - 'the visual or tactile surface characteristics and appearance of something'. This also remains a vague and generalized definition for any kind of study of this visual element. Surprisingly, even with frequent occurrence of visual textures as results or element of interest in various fields of study like shearography, fingerprinting and others, very little literature is actually available which deals with standardizing or defining of texture [53].

A general description that suits our basic purpose in this work, that of studying fingerprints and dealing with image enhancements related to them, is given by Rao [53]. It states tex-

CHAPTER 3. ORIENTED TEXTURE IMAGE PROCESSING

Figure 3.1: Comparison of different kinds of textures.

ture as 'the surface marking or two dimensional representation of a surface'. Fingerprints, since they are formed by the repeating ridge and valley structures on the surface of the fingertips, are well covered by this definition.

Figure 3.1 shows three different kinds of textures that can easily be defined by this. The first image is that of grains of sand. Second image is of rotten wood and the third is a Fingerprint. The immense variety of texture features available makes further classification necessary to make it more meaningful in our context. Orientation or direction is one property that has no relevance in case of the random distribution in figure 3.1(a) and 3.1(b). Here, the directions of the edges in the image shift abruptly and arbitrarily. In the image of the fingerprint (Figure 3.1(c)) however, the direction has a continuity. Within close adjacent neighborhoods for most part of the image the variation in direction is not extreme. A mapping of direction in this case, using software means, thus gives us a more useful information regarding the texture.

Several methods have been proposed to get the orientation information from the fingerprints [51, 54, 55]. These methods can generally be classified into gradient-based methods and filter based methods.

3.2 Orientation Detection

3.2.1 Gradient Based Orientation Detection

Gradient-based methods for detection of orientation are closely related to the edge detection techniques. The basic operator meant to gather information is the same Gradient operator used for detecting edges. These operators act on the intensity difference between neighboring pixels to give information of edge lines. In ideal case, this intensity difference can be taken as a step function though in real world it is hardly so. Fingerprint images in fact are supposed to have sinusoidal pattern for intensity distribution in a direction orthogonal to the predominant orientation of the neighborhood [34].

3.2.1.1 Edge Detection

Edge detection is a field commonly encountered in applications of machine vision and feature detection [56]. This is also the reason for a wide availability of research and literature on it. For a typical system involving any sort of automation using vision, specially in the segmentation and feature detection phase, the application of edge detection becomes more or less imperative. As mentioned before, an ideal model for an edge in the image is a step function that in terms of images will mean a change in the gray level between two regions. For the sake of understanding, it's better to look at a one-dimensional case and analyze it.

The change of intensity can be modeled as shown in figure 3.2. The region between the two dotted lines indicates the region of intensity change in the image. A black arrowhead points to the maximum rate of intensity change in the border region. A gradient equation can now be introduced.

$$\nabla I = [\partial I / \partial x, \partial I / \partial y]. \quad (3.1)$$

Where (x, y) are pixel coordinates and I is the intensity.

CHAPTER 3. ORIENTED TEXTURE IMAGE PROCESSING

Figure 3.2: Edge

Figure 3.3: Edge: First Differential

Taking the rate of change of intensity as indicated by Eqn 3.1, leads to the curve profile in figure 3.3. The black arrowhead shows the point of highest intensity change in the edge region. The figure again is a one dimensional representation of the two dimensional operation and hence in terms of the equation will represent either $\partial I / \partial x$ or $\partial I / \partial y$.

The value at the regions of no variation remains zero while the peak is detected at the region of greatest variation in the gradient results. This peak is essentially the location of the edge in the image. Mathematically, the peak can be detected in the change of values by using a second derivative or a derivative of the results shown in figure 3.3. This stems from the fact that the direction for the rate of change between the edge beginning (indicated by

CHAPTER 3. ORIENTED TEXTURE IMAGE PROCESSING

Figure 3.4: Edge: Second Differential

the dashed line on the left) and the peak position and that of the region from the peak position to the right hand side end (the dashed line at the right side) is opposite leading to a zero value at the actual peak (change in direction). The overall equation for the second derivative, also called the laplacian is given in Eqn 3.2.

$$\nabla^2 I = \partial^2 I / \partial x^2 + \partial^2 I / \partial y^2. \quad (3.2)$$

Again, sticking with the one dimensional representation of the equation or the $\partial^2 I / \partial x^2$ part, the zero crossing can be seen in figure 3.4 below -

The black arrowhead representing the highest rate of change of intensity in the original figure 3.2 is now sitting on a value zero hence marking the point of edge. The expressions shown till now represent the edge detection in a continuous or in other words 'Analog' domain. Image processing with the help of a computer requires dealing with a digitized expression or a two dimensional array of numbers representing the gray scale values at appropriate sampling rate. A simple gradient operation in this domain can be expressed as a pure subtraction operation. If $I(x,y)$ represents the value of grayscale at pixel defined by the co-ordinate (x,y) , then in the direction x , the expression can be shown by $I(x+1,y) - I(x,y)$. In fact various approximations of this expression form the different gradient masks.

CHAPTER 3. ORIENTED TEXTURE IMAGE PROCESSING

The simplest is the Roberts mask [57] and the mask for the directions 'x' and 'y' are given by -

$$\begin{array}{cc} +0 & +1 \\ -1 & +0 \\ \& \\ +1 & +0 \\ +0 & -1 \end{array}$$

Similarly, a set of 3x3 masks defined by 'Prewitt' operator [57] are defined as -

$$\begin{array}{ccc} -1 & +0 & +1 \\ -1 & +0 & +1 \\ -1 & +0 & +1 \\ \& \\ +1 & +1 & +1 \\ +0 & +0 & +0 \\ -1 & -1 & -1 \end{array}$$

Modification of the 'Prewitt' masks with more emphasis on the pixels in the four connected neighborhood defined in previous chapter leads to 'Sobel' [57] masks -

$$\begin{array}{ccc} -1 & +0 & +1 \\ -2 & +0 & +2 \\ -1 & +0 & +2 \\ \& \\ +1 & +2 & +1 \\ +0 & +0 & +0 \\ -1 & -2 & -1 \end{array}$$

The laplacian has been utilized by other edge detection techniques like the 'Marr and Hildreth' [58,59] and 'Canny' [59,60].

Using the independent masks and combining the results obtained for the two orthogonal directions can give the location of all the edge pixels. The operators in their basic form are highly sensitive to noise and to avoid this issue, edge operators like 'Canny' use a 'Gaussian' smoothing filter before operation by the differential equations. In case of fingerprints, the images of ridges form the edges with the alternating valley locations.

Once the gradient values from both the masks have been obtained, the overall strength of the gradient and the value for the direction can be obtained by use of geometric equations dealing with standard Cartesian co-ordinate systems. If G_x is the value obtained in the x

---

CHAPTER 3. ORIENTED TEXTURE IMAGE PROCESSING

---

direction and G_y is the value obtained in y direction. The value of overall strength is given by equation 3.3. The overall direction is given by equation 3.4.

$$\sqrt{G_x^2 + G_y^2} = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \quad (3.3)$$

$$\tan^{-1} \frac{G_x}{G_y} = \tan^{-1} \left(\frac{\partial I}{\partial x} / \frac{\partial I}{\partial y} \right) \quad (3.4)$$

Once the direction has been obtained, in order to get the dominant direction in a bigger window region, a vector sum of the orientation values of each pixel is calculated [31]. The window is normally chosen as non overlapping square regions in the image.

The process of edge detection, by itself, also makes use of the direction information in order to fill up the breaking points and connect various edge lines. This makes the edge detection process more robust to noise and thus makes it a more practical tool for application in real world situations.

3.2.2 Filter Based Methods for Orientation Detection

3.2.2.1 Rotating Kernel

Rotating kernel method to find the orientation of texture in an image is more akin to line fitting. Originally proposed as a purely optical setup that can have software based implementation that was computationally expensive, it's meant primarily for edge enhancement [61]. The nature of this work makes it essential to try wide variety of methods as orientation detection forms the main area of focus.

Lee and Rhodes [61] proposed a method to get the image information by filtering the input image by a straight line mask oriented at an angle. This angle can be changed at discrete steps or in a continuous fashion as their mask was mechanical in nature and inline with the optical assembly. It's a simple mask as shown in Figure 3.5.

CHAPTER 3. ORIENTED TEXTURE IMAGE PROCESSING

Figure 3.5: Rotating Kernel Mask

To get the output, several of these masks were used in an array fashion with size being determined by the size of the window set for edge enhancement on the output image. Taking the case of a single window, the output of the setup is obtained by equation 3.5.

$$S(x, y) = I(x, y) * K(x, y) \quad (3.5)$$

where $S(x, y)$ is the final output, $I(x, y)$ is the input image and $K(x, y)$ represents the mask oriented at an angle θ .

For each window with an edge line, two extreme readings are generated while rotating the mask. These are given by equation 3.6 and equation 3.7.

$$Max(x, y) = maximum\{S(x, y) : 0 \leq \theta < 360\}; \quad (3.6)$$

$$Min(x, y) = minimum\{S(x, y) : 0 \leq \theta < 360\}; \quad (3.7)$$

At an edge line between two regions, the difference between the maximum and minimum value will be high. This method had good results for the processing of cell boundaries. Going back to the textures shown in figure 3.1, it can be seen that images that are going to be dealt with, i.e the fingerprint images, have well defined lines separating regions and closely resemble the profile shown in figure 3.6(b).

CHAPTER 3. ORIENTED TEXTURE IMAGE PROCESSING

Figure 3.6: (a) Ideal edge as a solid line and edge as a slow change shown in dashed line.(b) Edge lines encountered in our case of textures.

6	5	4	3	2	
7	6	5	4	3	2
	7			1	
0	0			0	0
	1			7	
1	2	3	4	5	6
2	3	4	5	6	

Figure 3.7: Stock and Swonger Mask

The edge enhancement can now be achieved by using both the maximum and minimum values obtained in the convolved output for a pixel. The direction value for this is obtained as an output of a function that uses the difference between maximum and minimum grayscale value.

Next, another filter based orientation detection method that has been implemented in case of textures like fingerprints- 'the Stock and Swonger method' [51] is explained.

3.2.2.2 Stock and Swonger

In their paper on fingerprint classification, Karu and Jain use a filter-based method based on a 9x9 mask adapted from a paper by Stock and Swonger [51]. The mask is shown in figure 3.7.

CHAPTER 3. ORIENTED TEXTURE IMAGE PROCESSING

The values shown in the array are not filter mask values per se but are index numbers similar to the ones mentioned in the previous section on rotating kernel. These numbers indicate the neighbors of the pixel located at the center which constitute a certain direction. For example, looking at the position of pixels with index number '2' when seen relative to the center will constitute a line segment at 45° . Pixels indexed with 0 will constitute line segment at 0° and so on.

Sums are obtained for all of the directions based on the array above in order to get the orientation. This can be further illustrated by looking at equation 3.8 below -

$$S1 = I(i-2, j-4) + I(i-1, j-2) + I(i+1, j+2) + I(i+2, j+4) \quad (3.8)$$

$S1$ here is the sum according to all the neighbors marked with 1 in the mask. $I(i, j)$ indicates the value at pixel (i, j) . $S1$ as a result is also valid for pixel (i, j) . Once all the sums have been obtained for a pixel, i.e. $S0, S1, S2, S3, S4, S5, S6$ and $S7$, the minimum and maximum values are taken and depending on whether the center pixel is in a dark or light region, the minimum or maximum sum is assigned as the direction of the pixel.

What constitutes a dark or a light region needs to be clarified further. If $I(i, j)$ indicates the value at the center pixel then if the condition shown in equation 3.9 is met, the pixel is considered to be from dark region and is assigned the index of the minimum sum as the direction and that of the maximum sum otherwise.

$$4 * I(x, y) + \min(S) + \max(S) > 3/8 * \text{Sum}(S0, S1, S2, S3, S4, S5, S6, S7) . \quad (3.9)$$

Again, once the value of each pixel has been obtained, there is a need to average it out over a windowed region to reduce the effect of noise. This is done by taking the angle corresponding to each index and assigning unit vectors based on it to the pixels. A vector sum of these will lead to a vector in averaged direction for the window.

CHAPTER 3. ORIENTED TEXTURE IMAGE PROCESSING

The methods described above are the ones that have been concentrated upon in the work till now. Other methods of orientation detection do exist in the vast literature and some of them have been introduced in the next section. For the sake of organization, they haven't been classified into any specific group of methods.

3.2.3 Other Methods

3.2.3.1 Gorman and Nickerson

The method used by Gorman and Nickerson in their paper [43] on filter design is designed around taking special frequency of the image data in a perpendicular direction along three orientation angles - $0, \pi/3, 2\pi/3$. A circular average of these is calculated to factor in the circular nature of the orientation property (repeats at the interval of π) and is assigned as the dominant orientation of the region.

The method builds around the power equation of a one dimensional signal obtained from the line perpendicular to the orientation value under consideration (i.e. from the set of $\{0, \pi/2, 2\pi/3\}$). For the sake of simplicity, orientation angle of '0' is considered. The power equation for co-ordinate (x_0, y_0) is shown in Equation 3.10.

$$\text{Sum } x_a - x_b [\text{sum } y_a - y_b [I^2(x, y) - \{1/k(\text{sum } y_a - y_b [I(x, y)])\}^2] \quad (3.10)$$

where-

$$x_a = x_0 - (k - 1) / 2$$

$$x_b = x_0 + (k - 1) / 2$$

$$y_a = y_0 - (k - 1) / 2$$

$$y_b = y_0 + (k - 1) / 2$$

$k * k$ is the size of the window region.

$I(x, y)$ is pixel value at co-ordinate (x, y) .

CHAPTER 3. ORIENTED TEXTURE IMAGE PROCESSING

3.2.3.2 Mardia

Semi-variogram is another statistical tool which has been employed for gathering the orientation value of the ridges. In their paper in Pattern Recognition Letters(1997), K.V Mardia et al [62] propose the algorithms based around it. The value of semi-variogram is calculated at 16 equi-spaced orientations and a high frequency response can be seen as the direction becomes orthogonal to actual ridge direction.

Semi-variogram for orientation value of 0° is given by the Equation 3.11 [62].

$$\ddot{g}_y(h; 0) = \frac{1}{2(2r+1-h)} \sum_{k=-r}^{r-h} (y_k - y_{k+h}) \quad (3.11)$$

where -

h is the lag value limited by r ;

r is the radius;

y_k is the grayscale value at distance k from (i, j) ;

3.2.3.3 Mehtre

A reasoning similar to one proposed by Mardia is used in the concept of directional images for finding the orientation by Mehtre et al [54]. The method is represented by equation 3.12 and is calculated for multiple orientations [54]. Figure 3.8 give a clearer picture of the kind of calculation involved.

$$K(i, j) = \text{Min} \sum_{k=-1}^n [C(i_k, j_k) - C(i, j)] \quad (3.12)$$

where-

n is the number of pixels chosen for the computation;

$K(i, j)$ is the direction at the point;

$C(i, j)$ is the greyscale value at the point;

$C(i_k, j_k)$ is the extrapolation of $C(i, j)$ in the direction k .

CHAPTER 3. ORIENTED TEXTURE IMAGE PROCESSING

Figure 3.8: Orientation Image Calculation

3.3 Orientation Modeling

Orientation modeling based on the detection of the core points and sometimes as a composite of standard orientation detection is done to enhance the detected orientation field of the input image. One method proposed by Sherlock and Munroe [63] was to take core points as poles and delta points as zeros in a complex plane and use the distance parameter from their location to synthesize the fingerprint's orientation field.

An improvement of this method was proposed by Vizkaya and Gerhardt [64] who proposed a piecewise linear model which gave a better result for modeling though still not the perfect one. They described the earlier zero-pole model as a zeroth order approximation of the actual problem and thus only a step towards the solution that requires further study.

Later developments in the field has led to a model based method proposed by Zhou and Jinwei [65] who have attempted at fitting a polynomial function while giving appropriate weightage to regions near core and delta positions which have a higher curvature and thus a higher variation of orientation in the neighborhood.

The introduction to orientation modeling has been included in this chapter as it is one of the areas not directly covered by this research work but which can have an application in

CHAPTER 3. ORIENTED TEXTURE IMAGE PROCESSING

discriminating between overlapped fingerprints which are shown in chapter 6.

3.4 Quality Measure

Fingerprint orientation estimation methods mentioned above, excluding the modeling methods since they are a secondary process applied after the core and delta points have been obtained using coarse orientation detection methods, are susceptible to noise. Further, since the input image quality is very low in quite a few cases, for example latent prints, a quality measurement algorithm is required to determine the validity of orientation value detected. In a paper by En Zhu et al [66], a neural network output based on eleven quality parameters is used to achieve this. A high output of this network indicates a correctly identified orientation and a low output indicates a false detection.

$$C1 = \sqrt{\left(\sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} (G_x^2(u, v) - G_y^2(u, v)) \right)^2 + \left(\sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} (2 \cdot G_x^2(u, v) G_y^2(u, v)) \right)^2} \quad (3.13)$$

$$C2 = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} \sqrt{(G_x^2(u, v) - G_y^2(u, v))^2 + (2 \cdot G_x^2(u, v) G_y^2(u, v))^2} \quad (3.14)$$

In the course of this research work, two of these quality parameters, which are not dependant on detected orientation, are used and hence it is necessary to provide the equations for these in this chapter. Equations 3.13 and 3.14 describe these two parameters hereby termed $C1$ and $C2$.

Here G_x and G_y represent the same gradient operators as mentioned in section 3.2.1. u and v indicate the coordinates in the image and w is the window size. These quality measures use the strength of the gradient to determine if the data represented by a section of the image contains ridge details.

Another quality measure uses a coherence value instead of gradient strength. Coherence

CHAPTER 3. ORIENTED TEXTURE IMAGE PROCESSING

Figure 3.9: Gabor filter at 45° (a) A Gabor Filter Response. (b) Bandpass characteristics in orthogonal direction with Gaussian pattern in the direction of the filter.

value here refers to coherence present in the orientation values and not the coherence of the grayscale values in the windowed region of the input image. This parameter has been used in a paper by Yi Wang et al [67] and the relevant calculation required for this is shown by equation 3.15 where Coh represents the coherence score, $G_{sx}(u, v) = G_x^2(u, v) - G_y^2(u, v)$ and $G_{sy} = 2G_x(u, v)G_y(u, v)$.

$$Coh = \frac{|\sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} (G_{sx}(u, v), G_{sy}(u, v))|}{\sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} |(G_{sx}(u, v), G_{sy}(u, v))|} \quad (3.15)$$

3.5 Filter Design

3.5.1 Gabor Filters

The intensity values of ridge-valley structure of fingerprints, when sampled in an orthogonal direction to a ridge, closely resemble a sinusoid in a local neighborhood devoid of minutiae points [34]. Also, the frequency of ridge valley structure is almost constant [68]. This property has been used to enhance fingerprint images using a Gabor Filter [18,34,68].

Gabor filters have orientation and frequency specific properties as is shown in figure 3.9. These filters are applied to a fingerprint image to selectively enhance the ridges present in a specific direction. The frequency parameter for these filters is tuned based on the

---

CHAPTER 3. ORIENTED TEXTURE IMAGE PROCESSING

$$g(x, y) = \exp\left(\frac{-1}{2} \left[\frac{x_\phi^2}{\sigma_x^2} + \frac{y_\phi^2}{\sigma_y^2} \right]\right) \cos\left(\frac{2\pi x_\phi}{T}\right) \quad (3.16)$$

$$\begin{aligned} x_\phi &= x \cos \phi + y \sin \phi \\ y_\phi &= -x \sin \phi + y \cos \phi \end{aligned}$$

frequency value of the print. Since the frequency shift is minimal in a print and a fixed scanning resolution is normally used (500dpi in case of FBI standard), this can be set at a fixed value. For the angular resolution, a bank of eight filters is used together.

Two-dimensional representation for this filter is given by equation 3.16. The three tunable parameters T , σ_x and σ_y , apart from orientation θ can be determined experimentally based on best possible results. Further adjustable parameters can be introduced based on need as by Jianwei Yang et al [18].

3.5.2 Bandpass Filter

Gabor filters are used for their orientation and frequency specific properties. Also, they act like bandpass filters and selectively enhancing artifacts at a fixed frequency. If a traditional bandpass filter is expressed in polar coordinates and its angular and radial components are separated, it too finds an application in the field of fingerprint image enhancement.

With polar co-ordinates given by (ρ, φ) , the response of the filter is given in equation 3.17 [69].

$$H(\rho, \varphi) = H_{radial}(\rho) \cdot H_{angular}(\varphi) \quad (3.17)$$

An implementation of this filter with Butterworth approximation has been given in a chapter by Sherlock in the book “Automatic Fingerprint Recognition System” [22].

The next chapter deals with the experimental setup and the details of the software developed for this research work. It also mentions the various tools used for carrying out the

CHAPTER 3. ORIENTED TEXTURE IMAGE PROCESSING

---

image processing operations.

Chapter 4

Project Components and Software Development

Summary: During the course of this work different software tools, libraries and hardware modules were used. This chapter provides a general description of these in order to give an overall picture of the project. Later part of this chapter includes an introduction to the software finally developed as the outcome of this work.

4.1 Introduction

This research work was primarily software based though the initial start was related to a new hardware setup for detection of latent fingerprints employing the use of high frequency modulated laser and focusing on fluorescence detection by means of homodyne and heterodyne signal processing methods. To give the complete picture of the project and the work done, this chapter introduces the reader to all the tools used, the reason for their use and the relation between them.

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

4.2 Tools

The different components used in this project have been listed down under the headings 'Hardware' and 'Software' in section 4.2.1 and section 4.2.2 respectively.

4.2.1 Hardware

Two hardware components, apart from modules used in the earlier work dealing with the latent fingerprint setup, were used and their details have been listed below. Details of the latent print detection setup can be obtained from [70] [71] and [72]. A brief description of this is also given in the later part of this chapter in section 4.3.1.

- **Notebook**

- **Model:** HP Compaq nx8420
- **CPU:** Intel Centrino Core Duo T2300 1.66GHz
- **RAM:** 1GB at 980MHz
- **OS:** Microsoft Windows XP Professional
- **GPU:** ATI Mobility Radeon X1600

- **Scanner**

- **Model:** Worldcard Office Card Scanner
- **Max Colour Depth:** 256 colour grayscale
- **Max Scan Resolution:** 600 dpi

4.2.2 Software

The focus of this research was development of image processing algorithms which are more suitable for orientation detection in fingerprint images. That is, more suitable based

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

Figure 4.1: Microsoft Visual C++ .NET Express Edition

on comparison with current methodologies and specifically for our case concerning latent prints which may or may not have an oriented pattern in their background. The development environment, compiler, image and GUI libraries, quality measure and simulation softwares used for this purpose have been introduced in the sub-sections below.

4.2.2.1 Visual C++ .NET

C++ was chosen as the programming language for the project due to wide variety of free and efficient image processing and other libraries easily available for it. The flexibility and ease of implementing 'Object Oriented Programming' paradigms was also a contributing factor. For a research work, this becomes important as code readability and reuse are of paramount importance for a continuously evolving field of study.

Microsoft Visual C++ .Net 2005 was the choice for the 'Integrated Development Environment' or IDE. An IDE provides a convenient solution integrating code editors, code coloring and compiler for program output. The specific flavor of this product used was the 'Express Edition' which is a free version provided by Microsoft and comes with a very open license with no restrictions on use for any purpose including educational or commercial.

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

The 'Express Edition' is a lightweight version of the mainstream flavor but includes the essential libraries like GDI+, Windows Forms and Win32 for development on a 'Windows' operating system.

4.2.2.2 GDI+

GDI+ is a graphics library available with Microsoft Visual C++ 'Express' edition. It is a class based Application Programmers Interface (API) and provides support for manipulating picture, text and other drawing objects like geometric shapes. GDI+, as the name implies, is an evolution from an earlier library 'GDI' and acts as an interface between the Win32 (the standard windows library) and the graphical device drivers i.e. the video card drivers. This also implies that Win32 or for that matter Win64, the 64 bit version for Win32, based applications do not directly interact with the graphics device drivers. GDI+ is used extensively with the library mentioned in the next section, the 'Winforms' library.

4.2.2.3 Winforms

Winforms or 'Windows Forms' is an API provided in .NET Framework by Microsoft. It provides different windows graphical user interface elements for assisting in the development of windows based programs. Winforms is a managed C++ library which is capable of doing memory management without explicit intervention by the code written by the programmer and hence adds to the stability of the systems. The ease of use for this library can be compared favorably with the earlier Windows API based Microsoft Foundation Classes (MFC) and this contributes to rapid development of programs with more availability of programmer time for development of solutions rather than development of an interface.

4.2.2.4 OpenCV

The main engine of this project, the image processing library used for implementing filters and manipulating image data, is the open source 'OpenCV'. This library was originally de-

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

---

veloped by Intel and now is supported by open source community. Its free for commercial, educational and research use and is distributed under the Berkeley Standard Distribution (BSD) licence. This library has been optimized for image processing in real time much akin to different commercial libraries available for machine vision applications.

Functions provided by this library can be broken down, based on their utility, into following categories -

- Core
- Image Processing
- Graphical User Interface (GUI)
- Machine Learning
- Camera Interface

Core components are the data structures used by the OpenCV library for representing the information. These specify the headers, the matrix elements, the geometric objects and their formats. Image processing components provide a way of manipulating these data elements for example applying a specific filter to an image or changing the format of the image data from RGB to grayscale. GUI components provide tools for presentation of the data. Machine learning components constitute statistical tools and Camera Interface functions are useful in integrating the input modules to an OpenCV based program.

This project primarily uses Core and Image Processing components of this library. GUI components were useful for debugging of the program when the OpenCV library was still being integrated with the Win32 components as no direct conversion tool between the image formats used by Winforms and OpenCV were available. Once the integration was completed, these components were removed from the program.

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

4.2.2.5 MATLAB

MATLAB is an engineering tool dealing with numerical calculations. It is a high level language environment which provides an easy to use functions for carrying out computations on standard engineering problems. The basic data representation or data structure in MATLAB is a matrix. This tool is a popular program available in the universities and a script based development environment that it provides helps in rapid development of simulations and platforms for algorithm testing. Apart from MATLAB which is provided by Mathworks, similar numerical problem solving tools are provided by INRIA in its Scilab program which is Open Source and Octave.

After evaluation of all three, the initial part of developing standard fingerprint orientation detection algorithms was carried out on MATLAB due to much better toolset available with its 'Image Processing' and 'Signal Processing' toolkits.

4.2.2.6 Verifinger

For the purpose of comparison of some results, commercially available Verifinger Software Development Kit (SDK) was chosen. This SDK is available from Neurotechnologija and the latest version, which is used in this work, is version 5.0. Neurotechnologija website provides a sample program 'Fingerpoint' which makes use of this SDK. The matching algorithm is capable of displaying a matching score in this sample program and its results have been included in certain sections of the results presentation.

4.3 Project Work

Previous sections of this chapter described the tools used for carrying out the work. This section now describes the actual work carried out. Since, as mentioned earlier, this work started as a part of research group dealing with detection of latent prints, we first briefly

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

---

describe the detection setup and the contribution of the earlier part of this work to it. Later, we look at the main part of the work dealing with biometric software development. The new algorithms designed and used in this software are described separately along with their results in Chapter 5 and Chapter 6. For a structured approach to the presentation of the work done, it has been subdivided into three interlinked phases which are elaborated on below.

4.3.1 Phase 1: Fingerprint Detection

This is a short description of the setup for the 'Phase Resolved' detection technique. This setup made use of modulated laser light and an intensifier with modulated gain setting on the detection side to achieve a greater signal strength and hence a clearer image compared to conventional fingerprint detection techniques. Description of what exactly was involved in this is given in next section.

4.3.1.1 Hardware Interfacing

Figure 4.2 shows the two different setups used to carry out detection of fingerprints. Setup in figure 4.2 (a) uses 'Function Generator 1' ($FG1$) to modulate the 'Electro Optic Modulator' (EOM) which in turn modulates the intensity of the laser coming to the 'Liquid Guide' (LG). $FG1$ is hard connected to 'Function Generator 2' ($FG2$) for synchronisation. A synchronised signal from $FG2$ is thus sent to the 'High Rate Intensifier' (HRI) mounted in front of the CCD camera. This is referred to as the 'Homodyne' setup.

Figure 4.2 (b) shows the 'Heterodyne' setup. Here, the connections are basically the same as the homodyne setup of figure 4.2 (a) with a slight difference. The synchronization of the modulating signals for interrogation and detection, i.e. signals which come out of $FG1$ and $FG2$ respectively, has been modified and a low frequency oscillator now decides the frequency difference between these two. This slight difference in the frequency of the two signals is responsible for a higher Signal to Noise Ratio or SNR in this setup. Since this

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

Figure 4.2: Schematic for (a) homodyne and (b) heterodyne assisted phase resolved technique

is only an introduction and a greater depth into these is not needed for the purpose of this project, the theoretical formulations are not given. They can be obtained from [73].

4.3.1.2 Automation

The tuning parameters of the setup shown in figure 4.2 [73] are controlled by the *HRI*, *FG1* and *FG2*. All three of these decide the waveforms, the modulation frequency and exposure times which affect the quality of the image captured. Instead of using manual methods to perform an acquisition, a script program based on Davis software provided by LaVision was developed. The software script featured an easy to use GUI with GPIB bus

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

Figure 4.3: Sample Images obtained by phase resolved setups

based control of the three tuning components in order to have a single interface for control of the image capture.

For further automation, an auto capture script which steps through a frequency range and then assesses the image quality on each increment step was written. This script provides the best possible result from the setup. The quality measure was specifically designed for the images obtained i.e. fingerprints.

4.3.1.3 Images

Shown in figure 4.3 are the images obtained using the two phase resolved setups described above. This is a sample set of the results which will be used later in chapter 5 and chapter 6 for a comparative analysis of standard and new techniques of orientation detection.

4.3.2 Phase 2: Test Algorithm Development

One of the main areas of study for this project was enhancement of overlapped fingerprint images. Different algorithms for orientation detection were tried on this kind of print to decide on the most suitable method to achieve this. The test image which was used during this phase is shown in figure 4.4. These standard algorithms, which have been mentioned earlier in chapter 3, were developed using MATLAB. The intermediate results shown in

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

Figure 4.4: Original Test Image

this section are based on this test image.

4.3.2.1 Rotating Kernel: A Software Implementation

Since the implementation of the rotating kernel method mentioned here differs from the one in literature [61], it is essential to mention the method adopted before proceeding to the results of orientation detection.

A software implementation of this method is followed in order to reduce the hardware requirements to a minimum. Processing time, considering the final outcome of the work will have an application in extremely specialized cases, is not an issue. Hence a computationally expensive method can be used.

Consider the image shown in figure 4.5, this is a rough case of what will be encountered as input in the oriented texture image enhancement procedure.

Taking an appropriate mask size that can be determined empirically, approximate a straight line of fixed width and rotate it through 180 degrees at fixed step intervals based on a

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

Figure 4.5: A representation of a typical input case

Figure 4.6: Masks at three orientations

compromise between computational speed and resolution required. All the orientations obtained are then stored as a set of 2D arrays in binary to be used as masks on the image.

The effect of using these masks on the image now is equivalent to using an averaging filter with a directional biasing. Looking closely at the case of dark lines on a light background, it can be deduced that when this filter is aligned with the orientation of the lines, an average of low values pixels is obtained and hence a low value results as the output for that pixel. To make the picture clearer, figure 4.6 shows the mask in three orientations, its graphical representation and its binary values for the actual implementation.

The results from each of these masks are stored separately in a volume buffer, or in other words, a stacked array ordered according to the mask index. This indexing helps in the

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

Figure 4.7: Orientation detection process

next phase of processing when the pixels are interrogated based on the value they have in each of these results.

Index numbers of the result with least value is assigned to the output array as the most likely orientation for the line contained in that pixel. The final output is the image sized array containing a range of value corresponding to the number of masks used. The process has been illustrated in figure 4.7.

Since having individual orientation values for pixels leads to a lot of non uniformity in the orientation map of the image and with the priori knowledge that pixels in immediate neighborhood will have similar orientation if they don't constitute the core point in case of a fingerprint, the output is windowed and likely orientation values of each of these windowed regions are obtained.

Each window contains entries of most likely orientation present at pixel values. The final orientation assigned to the window is the index number that is encountered most number of times in it.

4.3.2.2 Results of standard algorithms

The first step and the main area of focus, that of orientation detection, can be divided into two major methods. One involving the filter based analysis and the other based on the

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

Figure 4.8: Orientation detection result for 'Stock and Swonger' filter

gradient-based orientation detection. Since the process to be developed has to be automated as far as possible, i.e it should involve minimum operator intervention, the output of the filtering stage and hence the final result, which is essentially selective enhancement in a particular orientation, will depend completely on this stage. The result for the 'Stock and Swonger' method [51] has been shown in figure 4.8.

Implementation of the 'Stock and Swonger' method was done in a manner similar to the 'Rotating Kernel' method. The reason behind this implementation is mentioned later in the section and concerns a peculiarity of filter based approach of this sort which can be useful in case of overlapped prints.

The straight lines indicate the orientation detected for a region of 16x16 pixels in the image. Orientation of each of these line segments is the same as the obtained results. Application of the rotating kernel algorithm on the same image leads to the result shown in figure 4.9.

Though no significant difference in the results is visible, on detailed analysis, with the orientation of the top print as the criteria, the result obtained from the rotating kernel was found to be better. The orientation detection in the background of the images, composed

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

Figure 4.9: Orientation detection result for 'Rotating Kernel' filter

of non-print area, is a result of the method in which the filters were applied and their difference does not constitute the analysis of these results.

The next algorithm tried was gradient based [22]. The gradient mask chosen for the result in figure 4.10 is the 'Sobel' mask.

Result obtained from this gradient method was found to be better with higher consistency in the orientations between the neighboring regions. Though this method is much more susceptible to noise, in our case the process of averaging in windowed region and clarity of the test print has reduced the effect of random noise.

While more consistent result is seen for the 'Sobel' method, there are distinct advantages to using filter based methods when applied in a manner suggested in the previous section. Since a rotating kernel algorithm in our application essentially gives a set of eight different images, analysis for two distinct orientations present in image regions becomes much easier. On two mask conditions, a local maximum or a local minimum response by the image is seen. The figure shown in 4.11 clarifies this point.

The ridges in the test image are grey on white background. In this case, a minimum

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

Figure 4.10: Orientation detection result for 'Sobel' mask

Figure 4.11: Response at a pixel to different masks of 'Rotating Kernel'

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

---

response to a filter orientation at position 3 corresponds to the ridge orientation. This is the absolute minimum position. A second minima, though, is visible at position 6 indicating a likely presence of second ridge in this window. This second ridge might be from the background print.

In this case, though the value at position 2 is lesser than that at 6, thus indicating a higher likelihood of orientation represented by 2, the second orientation is not taken as 2 because it falls in the immediate neighborhood of the main detected orientation i.e. 3. In other words, position 2 does not constitute a local minima.

As a conclusion of phase 2 of this work, it was decided to go for an approach which can give results of higher consistency like in gradient based method and at the same time provide a mechanism for detecting multiple orientations in the neighborhood of a pixel. Computational time required for a hybrid algorithm, apart from the need for a simpler mechanism for future development, was the reason for a non MATLAB implementation using C++ and windows based GUI which was developed in phase 3 of the project. This approach also helps in easier and faster obtaining of results for comparison as instead of trying individual scripts, a central recipe based application common in industrial automation processes can be adopted. The software developed was named BiometricOCV

4.3.3 Phase 3: BiometricOCV Software Development

The C++ based implementation of the enhancement and orientation detection algorithms has been done in the program BiometricOCV described here. The image processing library used for carrying out various image format manipulations and filter applications is OpenCV. The GUI is done using winforms library and the input image format is bitmap. The result output is limited to 32 bit bitmap due to GDI+ support issues.

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

Figure 4.12: Main Window

Figure 4.13: Function Panel

4.3.3.1 Software Description

Figure 4.12 shows the main startup screen of this program. The panel on the right hand bottom side is the main function panel which provides access to the 'Analysis' window and the 'Recipe' panel. This panel is shown in figure 4.13.

We first look at the recipe panel shown in figure 4.14. This panel pops up when a user presses the recipe button on the function panel. The program uses two types of recipes, 'Main' recipe shown in figure 4.15 and 'Analysis' recipe shown in figure 4.16. Both these recipes are accessed using the recipe panel. They can also be accessed directly using the XML file stored in the 'Recipes' directory of BiometricOCV program.

Analysis recipe is used to store the main process and visualization settings. These settings are described in section 4.3.3.3. Several analysis recipes for different image input scenarios can be set up using the Analysis recipe panel. This is necessary as the number of variables in the input images to be processed will vary in the actual deployment scenario.

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

Figure 4.14: Recipe Panel

Figure 4.15: Main Recipe

This concept is a borrowed one and is commonly seen in various industrial machine vision applications. The recipe selections and storage is done by an expert user. Which recipe is to be used is decided by the 'Main' recipe which is chosen by the operator.

In actual practice, only with further expansion of the software to include more options like hardware settings for image acquisition or centralised database access for image input, will this method of storage become important. The purpose of having a main recipe is to choose the appropriate recipes, based on a particular configuration, for all different modules in the same window. Once saved, getting the correct setting for the modules becomes easier as only the correct main recipe needs to be loaded. At present, the only module present is the 'Analysis Window' for fingerprint analysis.

The main recipe window automatically enumerates the list of all analysis recipes present in the system. The recipe to be used can be selected using this list as can be seen in figure 4.15. The last main recipe used is automatically loaded on startup of the program.

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

Figure 4.16: Analysis Recipe

Once the recipes have been loaded or modified, the user clicks on the 'Analyse' button on the function panel to open up the 'Analysis' window. This is the main interface which shows the results obtained from an operation. The input image can be loaded by using the 'File' menu. The user then obtains the results using the 'Get Orientation' button situated on the bottom right hand side of this window. Keeping in mind that this is a research based project, all intermediate results are also stored by the program and can be accessed using the 'Get Previous' << and 'Get Next' >> buttons situated at the center of the window.

4.3.3.2 Structure

This section provides an introduction to the program structure. The main purpose of doing this is to help any future enhancements to be carried out on this work. The main modules of this program are shown in figure 4.18. ModDandler module compiles into Modhandler.dll and the main purpose of having it is to provide hardware access routines to the program. OCVision module similarly compiles into OCVision.dll and contains the user defined image processing routines which cannot be found in the library or which can be

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

Figure 4.17: Analysis Window

Figure 4.18: Program Structure

defined more efficiently outside the main OpenCV library.

Main module used is the ModVision and this compiles to ModVision.dll. Main classes present in this module are listed in figure 4.18. The classes that are encircled in the dotted lines labeled namespace constitute the ModVision namespace. Namespace is a programming construct and for the purpose of this report it is not necessary to go into its definition. Individual class definitions are provided below -

- **BForm:** This is the base class for main window and is used by BiometricOCV.exe. The reason for this being defined in the ModVision module is to allow access to other user interface classes to the recipe panel. A pointer to which is set in this class.

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

- **BTrainControl:** The analysis window for carrying out the enhancement process is provided by this class. It is also meant to be the generic class for other operations like template training for any pattern matching operations. Whenever a new analysis window is opened, it takes the settings from the recipes again. The windows which are opened before the recipe changes continue to use old settings. An instance of this class always contains an instance of the Biometric class.
- **BVisionPanel:** The main panel which provides access to the recipe settings and analysis window is contained in this class.
- **UIListRcp:** This is the main recipe panel providing access to analysis and main recipes. Since the pointer to this class is initialised right at the start of the main window and is referenced by it, the same instance of this class persists and is not cleaned up till the end of the program.
- **UIRcpAnalyse:** The user interface class for analysis recipe. This class provides access to all the settings needed by the Biometric class for carrying out the image processing operations.
- **UIRcpMain:** The user interface class for main recipe.
- **Biometrics:** This is the main engine for carrying out all operations on the input image. A pointer to the recipe instance, i.e. AnalyseRcp, is set to it on initialisation by the BTrainControl or a class derived from BTrainControl.
- **AnalyseRcp:** The main recipe class containing the settings to be used by Biometrics class. It also serializes the data to an XML file whenever it is saved.
- **MainRcp:** The recipe class to store settings for the main recipe. The data is serialized into an XML file.

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

4.3.3.3 Usage: Recipe Settings

This section will provide details of the settings available to the user in the analysis recipe shown in figure 4.16 (a) and (b). All the settings available have been classified under three groups -

1. Preprocessing
2. Orientation Detection
3. General

Preprocessing

Preprocessing refers to operations carried out on the image before the actual parameter gathering or enhancement routines. These operations tend to be global in character and their parameters are not tuned according to local information available. The parameter gathering routines in our case are those concerned with *Orientation Detection*.

Two preprocessing options are available, *Gaussian Smoothing* and *Histogram Equalization*. Gaussian smoothing is a standard smoothing operation which convolves the image with a gaussian mask. Tuning parameters, except for the mask size, are not available to the user. A larger mask size, which can be set using the *Mask Size* setting, will cause the image to blur more. Gaussian mask is used to filter noise from the image and is a standard preprocessing algorithm for edge detection.

Histogram Equalization is another global operation which has the primary aim of manipulating the pixel grayscale values in such a way that the pixel densities for all possible grayscale values in the image become equal, or as close to each other as possible. This has the effect of enhancing the contrast between different grayscale value regions of the image.

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

Orientation Detection

This group has the different options for orientation detection. *Window Size* is a common setting for all the methods and refers to a square window size in pixels which the input image will be divided into in order to calculate the orientation values. Since most images in sample are taken at around 600 dpi, a window size of 15 is the default value.

The next option available is *Use Filter Based Method*. Choosing this option leads to orientation detection either by the *Rotating Kernel* method if the *Stock and Swonger* option is not chosen or *Stock and Swonger* method otherwise. The filter sizes are fixed internally and only images with 500 dpi resolution can be used for these methods.

The gradient based method is applied when none of the three *Use Filter Based Method*, *Use Hybrid* and *Use Hybrid 2* options are chosen. The tuning parameters for the gradient based method are *Sobel Mask Size*, *Use Coherence* and *Coherence Threshold*. *Sobel Mask Size* parameter decides the size of the sobel mask which will be applied on to the image. Details of the sobel operator have already been discussed in chapter 3. *Use Coherence* option uses a coherence value to determine the print quality based on all the detected orientations in the window region. Used in [67], this parameter has been defined in chapter 3 section 3.4. A coherence score for a window varies from 0 indicating no coherence to 1 implying high coherence with direction detected for each pixel of the window being the same. The *Coherence Threshold* parameter is used to eliminate image window regions with a coherence value lower than the value set.

Use Hybrid and *Use Hybrid 2* are the newly developed orientation detection methods which use a quality measure to detect orientation of a window region. *Use Hybrid* is an intermediate algorithm before the development of *Use Hybrid 2* method and does not give good results. It has been preserved as an option only due to the fact that the algorithm structure for it is exactly the same as *Use Hybrid 2*. *Use Hybrid 2* method has been described in detail in chapter 5. Tuning parameters effective on the *Use Hybrid 2* method

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

---

are *Sobel Mask Size* and *Hybrid 2 Discriminator*. The quality measure is gradient based hence the importance of *Sobel Mask Size* parameter. It is also the discriminator and the equations for the two choices *C1* and *C2* have been provided in chapter 3 and [66].

General

Only three options in this group are valid for all the orientation detection methods mentioned in the previous group. These are the *Show Grid* the *Display Filtered Output* and the *Use Voting* option. *Voting Threshold* and *Continuity Threshold* are the parameter setting for the *Use Voting* option.

Show Grid option puts a grid structure based on the window size value on all the orientation detection and filtered output results. This grid helps in analysing the faults of a particular algorithm and relative advantages of other by providing an easier reference.

The *Display Filtered Output* option produces a filtered output result of the orientation detection. This filtered output is obtained by patching together the results of motion filtering and is provided just as a visual indication of what the filtered image will look like. It is not a filter designed for the actual purpose of fingerprint enhancement like the 'Gabor' or tuned 'Bandpass' but has characteristics inferior but similar to them.

Use Voting forces the use of a voting algorithm on the detected orientation map. It is an eight connected neighborhood based operation which has been described in chapter 2. Based on the threshold, the directions in this neighborhood of a window are queried and for each direction found which is within the threshold deviation from this window, a vote is added. Windows with a higher vote have a higher certainty of correct orientation value. This method has also been used as a quality measurement tool for different algorithms as it suppresses the window regions of low certainty or 'votes' based on the other tuning parameter the *Continuity Threshold*.

The remaining two parameters, *Use Second Pass* and *Use Vertical Scanning* are specific

CHAPTER 4. PROJECT COMPONENTS AND SOFTWARE DEVELOPMENT

---

only to the *Use Hybrid 2* method of orientation detection due to the peculiarity of the results it obtains. The *Use Second Pass* method has been described in detail in chapter 6. *Use Vertical Scanning* option just changes the direction of the scan as the *Use Second Pass* algorithm was found to be slightly biased towards the scanning direction.

The next two chapters, chapter 5 and 6 are a continuation of the work done. Since two new related methodologies were developed during the course of the software development process, which showed better results compared with standard methods mentioned in literature, the result presentation has been separated from this main chapter on the project work. All the results were obtained using the software described in this chapter.

Chapter 5

A novel orientation detection method based on quality measure

Summary: Orientation detection is one of the most crucial steps in the process of fingerprint image enhancement and consequently its registration and verification. Gradient based methods are the most common implementation for detecting the orientation map of a print. In this chapter, a new method for detection is proposed which has consistently shown better results when compared with standard gradient based approach. The proposed method, though based on gradient detection, does not make use of the vector sum calculation.

5.1 Introduction

Fingerprints have been used in extremely wide variety of fields. From replacing or augmenting password based authentication system to their use in the field of criminal investigations as a forensic tool, they have consistently shown to be of immense importance whenever establishing identity of a person was a critical issue. Availability of cheap fingerprint sensors in the market recently has caused their use to spread out and now fingerprint readers can be easily found even on consumer electronic devices like 'Laptops' and 'PDAs'.

CHAPTER 5. A NOVEL ORIENTATION DETECTION METHOD BASED ON QUALITY MEASURE

The widespread use of fingerprints in commercial field has shifted the focus of development to faster algorithms which can match a print input to the templates stored in a database in a matter of milliseconds. This requirement for speed thus transcends all the processes to be carried out on the input. Though many methods have been proposed on the matching algorithm, minutiae based ones are the most common. A typical flow of steps involved in the matching phase is shown in figure 5.1.

Of the steps mentioned, any speedup will be focused on the *Orientation Detection*, *Matching* or the *Enhancement* stage as these afford the maximum possible variation in approach. Gradient based methods are typically faster than the other approach, which is filter based, for the orientation detection stage. They do have a flaw in terms of sensitivity to noise but preprocessing and averaging out over windowed regions are normally enough to overcome this.

In certain cases, speed is not the primary criteria. An *Automatic Fingerprint Recognition System* or AFRS for example will have two distinct input conditions. The first condition involves the building up of the database or *Registration*. The other condition is that the input fingerprint has to be matched against this database for finding the person it belongs to. This is termed *Identification*. The difference in processing condition, though both *Registration* and *Identification* will go through the steps before matching stage shown in figure 5.1, is the quality of the input print. *Registration* process has high quality print as the input while *Identification* has low quality prints, normally, as the input. This is specially true for institutions like the police departments or the FBI which often have latent prints as the input for *Identification*.

Latent prints are obtained from a crime scene where they have been involuntarily left behind by a suspect. This involuntary nature, coupled with the fact that the surface on which they were found can have huge variation, ensures that they are typically of very low quality. In such a case, reliability instead of speed becomes more important.

CHAPTER 5. A NOVEL ORIENTATION DETECTION METHOD BASED ON QUALITY MEASURE

Figure 5.1: Fingerprint Matching

5.2 Orientation Detection

Before proceeding to the description of the new method, a short look at the current gradient based method is in order. A gradient based method obtains the filtered results of the image using two masks. One for the horizontal direction and the other for the vertical direction. Figure 5.2 shows the results of such an operation.

The gradient image for horizontal or x direction is shown in figure 5.2 (b) and the gradient image for vertical or y direction is shown in figure 5.2 (c). Input image is shown in figure 5.2 (a) and it is an 8 bit grayscale image. If for a pixel at coordinates (u, v) in input image I , the gradient value for x direction is given by $G_x(u, v)$ in x direction image I_x and gradient value for y direction is given by $G_y(u, v)$ in y direction image I_y , then a vector can be formed representing the magnitude and direction of the gradient value as $G_x + iG_y$.

With care given to the fact that vectors which are π radians apart will tend to cancel each other, which is not desirable in our case, the overall direction of a window in this input image can be found out by calculating a vector sum of all gradient vectors associated with the pixels in the region.

CHAPTER 5. A NOVEL ORIENTATION DETECTION METHOD BASED ON QUALITY MEASURE

Figure 5.2: Gradient detection

5.3 Proposed Method

The overall process of orientation detection and image filtering for the standard gradient method mentioned in the previous section and the one to be proposed in this one is shown in figure 5.3 (a) and (b) respectively.

The new method, as can be seen from figure 5.3 (b) first carries out the filterbank operation and generates the filtered images in eight discrete directions. The filter used for this work is a motion filter. It then uses the gradient operator on these to generate two sets of gradient results in x and y directions respectively. In the next step, a quality measure based on the gradient values is calculated for a windowed region. Two different quality measures, termed $C1$ and $C2$ here, were tried with similar results. The equations for these is given in equation 5.1 and equation 5.2 which have already been mentioned in chapter 3. The window size in the equations is considered to be w . (i, j) is the central pixel co-ordinate of the windowed region for which the quality is being calculated.

$$C1 = \sqrt{\left(\sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} (G_x^2(u, v) - G_y^2(u, v)) \right)^2 + \left(\sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} (2 \cdot G_x^2(u, v) G_y^2(u, v)) \right)^2} \quad (5.1)$$

$$C2 = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} \sqrt{(G_x^2(u, v) - G_y^2(u, v))^2 + (2 \cdot G_x^2(u, v) G_y^2(u, v))^2} \quad (5.2)$$

CHAPTER 5. A NOVEL ORIENTATION DETECTION METHOD BASED ON QUALITY MEASURE

Figure 5.3: Process flow of orientation detection methods

Once the quality for each of the eight filtered outputs of the filterbank has been calculated, a comparison is made for the filter response with best possible quality. The direction of this filter is then taken as the orientation of that windowed region.

5.4 Orientation Filtering

Once the orientation has been detected, regions or windows with false detections have to be eliminated from the output. We use a voting criteria with threshold for a vote set at 0 and threshold for elimination set at 3. In this voting scheme, each window checks its 8 neighboring windows for their direction. If the difference in direction of a neighbour is within the voting threshold, the window gets a vote incremented. In our case, since the threshold is zero, the direction of the neighbour should be the same as the window for it to get a vote. The elimination threshold decides the number of votes a window should have if it is to be judged as having the minimum required quality.

CHAPTER 5. A NOVEL ORIENTATION DETECTION METHOD BASED ON QUALITY MEASURE

5.5 Experimental Results

The results of this proposed method are shown in figure 5.4. The fingerprints have been obtained in-house using a 600 dpi card scanner. These prints are of relatively high quality. On visual inspection, it can be seen that all three prints exhibit a larger recovered area when they use the proposed algorithm.

To further confirm the results, 'Fingerprint version 1.4' program provided by Neurotechnology was used. Original prints were enrolled in the database and matching was performed on these prints. The matching scores have been shown below -

Fingerprint	Score: Standard	Score: Proposed
(a)	88	171
(b)	81	91
(c)	147	153

The algorithm was then tried on the fingerprints obtained from the phase-resolved technique mentioned in chapter 4. This was done to get an indication on the performance when low quality prints are encountered. The results of the standard and proposed algorithm for this are shown in figure 5.5. As can be seen, the difference in areas recovered is significant indicating that the new orientation detection technique does have an advantage in terms of quality than the standard gradient detection method.

5.6 Conclusion

In this chapter a new orientation detection approach for fingerprint images was illustrated. The results obtained do show an improvement over the standard gradient based method. Computationally, the new method is expensive both in terms of memory requirement and number of calculations to be carried out. But the advantages in terms of uniformity of orientation detected, as was shown by the voting criteria and quality thresholds, will outweigh the drawbacks in specialised cases where speed and memory footprint are not a concern.

CHAPTER 5. A NOVEL ORIENTATION DETECTION METHOD BASED ON QUALITY MEASURE

Figure 5.4: Results of the standard and the proposed method

CHAPTER 5. A NOVEL ORIENTATION DETECTION METHOD BASED ON QUALITY MEASURE

Figure 5.5: Result of the standard and the proposed method on phase resolved images

Chapter 6

A scheme for detecting multiple orientations in an image

Summary: A region in image with oriented texture can have multiple orientations. Standard gradient and filter based methods provide a single measure output indicating a single direction for any given windowed region of the image. Hence significant information is lost. Presence of noise is also a factor that can affect gradient based methods leading them to respond in a wrong manner and returning an erroneous result. In this chapter we propose a different approach to the issue of orientation detection and illustrate a method which can use multiple orientation values to provide better results than standard algorithms

6.1 Introduction

Noise elements, presence of a background artifact or presence of multiple overlapping prints in an area are possible conditions where a standard orientation detection technique won't give desired results. For such conditions a more robust method is required which is designed with these conditions in mind.

In chapter 5 we introduced a new 'quality based' method of orientation detection. This was shown to have better results when compared with the standard gradient based method. The method presented in this chapter exploits a peculiarity of the results obtained from the

CHAPTER 6. A SCHEME FOR DETECTING MULTIPLE ORIENTATIONS IN AN IMAGE

Figure 6.1: Two pass second maxima querying method for coarse orientation map

quality based algorithm to further improve the results obtained and to make the method more resilient to the presence of strong oriented patterns in the background.

6.2 Second Maxima

This peculiarity is that of being able to detect a second maxima or in other words, a second direction. In chapter 4 it was mentioned that the implementation of filter based methods, which was specific to this study, provided the results in terms of scores for each direction. Quality based method provides the results in a similar manner. For more details on the second maxima, see chapter 4 section 4.3.2.2. In order to make use of this property, a discriminator was constructed based on the voting criteria.

6.3 Discriminator and Direction Sensitivity

In chapter 5, assessment of quality was done using a voting criteria. To make a more efficient use of the results from the quality based method, a two pass method which queries the voting results on the maxima and second maxima value of direction found for a window region was used as the discriminator. The direction chosen was one which got more votes. This method is illustrated in figure 6.1.

CHAPTER 6. A SCHEME FOR DETECTING MULTIPLE ORIENTATIONS IN AN IMAGE

A property of this implementation is that it tends to be sensitive to the direction of querying when regions with multiple orientation form a significant part of the input image. This is because the votes received by an orientation value of a region will depend on the regions which have already been queried. This bias based on direction of query can be put to use in selectively enhancing a particular print.

Results of the second pass algorithm employing results from the quality based orientation detection method have been shown in the next section.

6.4 Results

The first set of results deal with strong oriented artifacts in the background. These have been shown in figure 6.2. Since the second pass algorithm has been specially tailored for this kind of input images, the orientation detection results are better when compared to the standard or quality based methods. To make these results clearer, figure 6.3 shows the results of filtering using these orientation maps as the input and keeping the quality threshold at three votes. The area to focus on, here, is the region right next to the background lines in the print. Standard and quality based methods show complete breakdown here as the gradient values contributed by this background element is much stronger than the print. The second pass algorithm, on the other hand, depends on votes apart from quality calculated on gradient values. This allows it to try the second maxima value which in this case is the correct orientation of the print. This second maxima value gets a higher vote and hence is chosen over the orientation output given by the first maxima.

Next case to be tried using second pass method were low quality prints which were obtained from the phase resolved method. Comparative results for these based on standard and quality based methods have been shown in chapter 5. Here we show the results from the quality based and second pass method in figure 6.4.

Again, the second pass method shows more robustness to noise elements and the additional

CHAPTER 6. A SCHEME FOR DETECTING MULTIPLE ORIENTATIONS IN AN IMAGE

Figure 6.2: Coarse orientation map output of the three methods

CHAPTER 6. A SCHEME FOR DETECTING MULTIPLE ORIENTATIONS IN AN IMAGE

Figure 6.3: Indicative filter output of the three methods

CHAPTER 6. A SCHEME FOR DETECTING MULTIPLE ORIENTATIONS IN AN IMAGE

Figure 6.4: Results of the quality based and the second pass method on phase resolved images

CHAPTER 6. A SCHEME FOR DETECTING MULTIPLE ORIENTATIONS IN AN IMAGE

---

recovered regions of these images have their orientations contributed to by the second maxima value of the quality based algorithm.

The third case of images we try this algorithm on are the overlapping prints. Overlapping prints have the largest area, percentage wise, with multiple orientation detections. Our implementation of double pass algorithm is not specifically tuned for this case but coarse orientation detection results obtained on two overlapping cases are given in figure 6.5. Their indicative filtered outputs are given in figure 6.6.

On these prints, it was observed that the print obtained later tended to have a higher degree of contribution to the orientation detected. This condition may not be true for all cases of overlapped prints but it is significant in the results obtained here.

Upon looking at the results, it can be seen that both quality based algorithms have significantly larger areas of print obtained based on our voting based quality measure when compared to the standard method. The difference in areas between these two, though, is not as big. This can be attributed to the observation made earlier that the top print is much stronger for both these cases and hence second maxima value is being used only in very few regions.

6.5 Conclusion

This chapter introduced a new method which takes orientation values as a quality measure thus making it possible to interrogate multiple orientations in a region. The key to effective use of the results obtained from a quality based algorithm is the discriminator which decides which level of maxima value to choose as the desired orientation in the region. A voting method was used here as the discriminator and it shows improvement in coarse orientation detection step for a fingerprint enhancement process. It also helps in illustrating a pathway to further reduce operator intervention in the enhancement process of fingerprints which had been mentioned in chapter 1.

CHAPTER 6. A SCHEME FOR DETECTING MULTIPLE ORIENTATIONS IN AN IMAGE

Figure 6.5: Coarse orientation results of standard, quality based and second pass algorithm on overlapped prints

CHAPTER 6. A SCHEME FOR DETECTING MULTIPLE ORIENTATIONS IN AN IMAGE

Figure 6.6: Indicative filtered results of standard, quality based and second pass algorithm on overlapped prints

Chapter 7

Conclusion and roadmap for future development

Summary: This chapter provides additional discussion related to the work done and primarily aims to highlight a future scope for this project. It details the potential areas for further improvement relating to the different development and algorithm design tasks carried out during the course of this study. Additional analysis of the results has been done here and known weaknesses with their possible remedies have been pointed out.

7.1 Introduction

Details of the work done for the completion of this work have been provided in the preceding chapters. Chapter 1, chapter 2 and chapter 3 provided the details of the relevant field concerning the science of fingerprints and oriented texture analysis. These chapters also introduced to the reader various tools that are used in carrying out the basic processes of classification, identification, enhancement, automation and such that are relevant to this field. Chapter 4 provided the details of the hardware, softwares and the libraries required for this study and for the development of the final biometric software. It also presented the results of the initial tests carried out at the beginning of the project and gave details of the software developed. Chapter 5 and chapter 6 contained the results of the new algorithmic tools developed as the result of the work carried out. The analysis of the results obtained

---

CHAPTER 7. CONCLUSION AND ROADMAP FOR FUTURE DEVELOPMENT

---

were discussed in the related chapters i.e. the chapters in which they were presented. This chapter provides a further analysis and looks at the areas of the work which can benefit from future research.

7.2 Quality Based Orientation Detection

The quality based orientation detection scheme was introduced in chapter 5. This orientation scheme utilises the results of the filterbank of eight motion filters to carry out the task of detection. Since a quality parameter in the normal enhancement and orientation detection scheme is used near the end of the process in order to seclude low quality areas, using it at the initial phase as done by the quality based scheme provides certain advantages. In a normal process, the results of a filterbank of eight direction specific enhancement filters will be used separately from the orientation detection mechanism in order to give a final filtered output. The orientation detected will be filtered before using quality parameters obtained from the input image and areas of quality lesser than the threshold will be left out. In the new scheme however, the quality enhancement process of filterbank operation is carried out before and inline with the orientation detection process. This will lead to more areas of higher quality to be detected as now the orientation detection process is looking at the final output quality instead of looking at the raw input image quality.

In chapter 5 this new method did give consistently better results with respect to our quality criteria. This criteria required orientations to be consistent within the neighborhood. Since the focus of this work was the detection of raw orientation maps and comparison between standard and newly developed detection techniques for the same, not much attention was given to the area of filter design. Bank of eight motion filters were used to generate faster results instead of using standard filter designs meant for fingerprint image enhancement. A possible method of improvement on the quality based method thus is use of standard fingerprint filters in the initial stage. This will provide inputs of higher quality to the orientation detection scheme. This improvement is illustrated in figure 7.1 which is a

CHAPTER 7. CONCLUSION AND ROADMAP FOR FUTURE DEVELOPMENT

Figure 7.1: Process improvement of quality based orientation detection

modified version of the figure 5.3 shown in chapter 5.

Since according to some literature, the ridge frequency of the map cannot be taken as a constant, the filtered images used in the initial stage can be taken as results from fixed frequency filters based on average frequency or the most likely frequency and then the orientation results can be used on tuned filter outputs for final results.

7.3 Second Pass Algorithm

Referring to the same figure 7.1, we can see that major improvement is possible on the *Second Pass Algorithm*. Current scheme utilises the multiple orientation detection capability of the quality based algorithm as mentioned in chapter 6. The proposed use of this feature is to make the detection more robust when strongly oriented features are present in the background of a larger print. i.e. the size of the fingerprint is much larger than the oriented noise features. This makes use of the continuity criteria in a neighborhood to determine whether the orientation detected as first maxima of the data set of eight is the

---

CHAPTER 7. CONCLUSION AND ROADMAP FOR FUTURE DEVELOPMENT

---

main orientation or if it is the second maxima. An enhancement of this method will be to use a more global approach. The orientation detected by the first and the second maxima can constitute two separate maps. The first maxima map will be the same as detected by the quality based algorithm without second pass. The second orientation map will contain the second maxima based orientations. A discriminator can then be used to form more complete set of two maps querying for continuity based on results from each. In the figure this discriminator has been termed as the *Best Fit Algorithm*.

The advantage of this approach is the possibility of a more enhanced main print in the case of overlapped prints. A partial recovery of the bottom print can thus also be looked into. Development of tools which can be used to effectively enhance overlapped prints was one of the areas of focus of this work and it was the reason why a multiple orientation detection capability was developed as the end result of this work. The suggested scheme provides an approach that can help in this case.

7.4 Conclusion

Apart from the development of two algorithms the implementations and results of which have been discussed in the previous sections, a stand alone software program named BiometricOCV was developed as the end result of this work. This software, which is completely based on free tools, is meant to be a convenient entry point for any other work to be carried out in the field of fingerprint enhancement. The complete list of the Open Source/ Free libraries used has been provided in chapter 4 and the use of these makes the development process a more flexible one. This software also works like a framework to be used for an integrated product level solution which can be provided to agencies dealing with latent prints.

A few significant omissions in the program are the orientation smoothing algorithms and the standard filtering methods. Standard filtering methods were already mentioned in the

CHAPTER 7. CONCLUSION AND ROADMAP FOR FUTURE DEVELOPMENT

---

previous sections. Orientation smoothing algorithms refer to the secondary process carried out on the coarse orientation maps to make them more consistent in their flow. They constitute one of the standard steps in any commercial application which deals with fingerprint image enhancement.

Bibliography

- [1] A. Ross, A. K. Jain, and J. Reisman, "A hybrid fingerprint matcher," *Pattern Recognition*, vol. 36, no. 7, pp. 1661–1673, 2003.
- [2] *State of Washington vs. Todd Ryan Todd*, 2000.
- [3] J. Balazs, "Federal rule of criminal procedure 29: A nuts and bolts guide to judgments of acquittal for criminal defence attorneys," *The Liberty Legend*, vol. 3, no. 4, 2006-07. web reference.
- [4] M. Darok, B. Reichenpfader, and B. Roll, "Finding a skeleton in altaussee lake - a forensic odyssey," *Forensic Science International*, vol. 147, pp. S45–S47, 2005.
- [5] S. J. Walsh, "Legal perceptions of forensic dna profiling," *Forensic Science International*, vol. 155, no. 1, pp. 51–60, 2004. Part 1: A review Of Legal Literature.
- [6] V. Causin, S. Schiavone, A. Marigo, and P. Carresi, "Bayesian framework for the evaluation of fiber evidence in a double murder - a case report," *Forensic Science International*, vol. 141, no. 3, pp. 159–170, 2004.
- [7] N. Paulsson and B. Stocklassa, "A real time color image processing systemfor forensic fiber investigations," *Forensic Science International*, vol. 103, no. 1, pp. 37–59, 1999.
- [8] *United States of America vs. Carlos Ivan Llera Plaza, Wilfredo Martinez Acosta and Victor Rodriguez*, 2002.

BIBLIOGRAPHY

- [9] S. Pankanti, S. Prabhakar, and A. K. Jain, "On the individuality of fingerprint," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 24, no. 8, pp. 1010–1025, 2002.
- [10] D. L. Grieve, "Daubert and fingerprints: The United States of America vs Bryan C. Mitchell," *The Sleuth*, April-June, 1999.
- [11] J. L. Mnookin, "A blow to the credibility of fingerprint evidence," *The Boston Globe*, February 2, 2004.
- [12] C. Stetkiewicz, "U.S. releases man questioned for madrid bombings," *Reuters*, May 21, 2004.
- [13] "Fingerprint analysis - the basics." <http://www.crimtrac.gov.au>.
- [14] R. Epstein, "Fingerprints meet daubert: The myth of fingerprint 'science' revealed," *Southern California Law Review*, vol. 75, 2002.
- [15] "Fingerprint trials." <http://onin.com>.
- [16] D. Maltoni and R. Cappelli, "Advances in fingerprint modeling." Article in Press, 2006.
- [17] A. Ross, S. C. Dass, and A. K. Jain, "Fingerprint warping using ridge curve correspondences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, January 2006.
- [18] J. Yang, L. Liu, T. Jiang, and Y. Fan, "A modified gabor filter design method for fingerprint image enhancement," *Pattern Recognition Letters*, vol. 24, 2003.
- [19] L. J. Sweet, "Man wrongly accused by fingerprints now faces drug rap," *Dactylogram*, vol. 1, no. 11, November, 2006. web reference.
- [20] J. P. Nielson, "Rebutting the 'no fingerprint' defense," *The Detail*, no. 70, 2002. web reference: <http://www.clpex.com/Articles/TheDetail/detailarchives.htm>.

BIBLIOGRAPHY

- [21] “Imaquest fingerprint enhancement software.” www.crimescope.com.
- [22] N. Ratha and R. Bolle, *Automatic Fingerprint Recognition Systems*. Springer, 2004.
- [23] *The Science Of Fingerprints*. GPO, Washington, D.C.: Federal Bureau of Investigation, 1984.
- [24] L. C. Jain, S. B. Lee, and I. Hayashi, *Intelligent Biometric Techniques in Fingerprint And Face Recognition*. CRC Press, 1999.
- [25] F. Song, X. An, J. Song, P. Ji, C. Bai, T. Lin, S. Li, B. Liu, and H. Wang, “Fingerprint identification by using a joint-fourier correlator and a liquid crystal light valve,” *Optics And Laser Technology*, vol. 26, no. 1, 1994.
- [26] X. Tan and B. Bhanu, “A robust two step approach for fingerprint identification,” *Pattern Recognition Letters*, vol. 24, pp. 2121–2134, 2003.
- [27] Y. He, J. Tian, X. Luo, and T. Zhang, “Image enhancement and minutiae matching in fingerprint verification,” *Pattern Recognition Letters*, vol. 24, pp. 1349 – 1360, 2003.
- [28] A. Jain, L. Hong, S. Pankanti, and R. Bole, “An identity authentication system using fingerprints,” *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1365–1388, Sept, 1997.
- [29] M. Tico and P. Kuosmanen, “Fingerprint matching using an orientation-based minutia descriptor,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, August 2003.
- [30] C. H. Park and H. Park, “Fingerprint classification using fast fourier transform and nonlinear discriminant analysis,” *Pattern Recognition*, vol. 38, pp. 495 – 503, 2005.
- [31] A. Jain and S. Pankanti, *Advances in Fingerprint Technology, 2nd ed*, ch. Automated Fingerprint Identification And Imaging Systems. Elsevier Science, 2001.
- [32] S. Wang, W. W. Zhang, and Y. S. Wang, “Fingerprint classification by directional fields,” *Fourth IEEE International Conference On Multimodal Interfaces (ICMI’02)*, October, 2002.

BIBLIOGRAPHY

- [33] P. Vizcaya and L. Gerhardt, "A nonlinear orientation model for global description of fingerprints," *Pattern Recognition*, vol. 29, no. 7, pp. 1221–1231, 1996.
- [34] L. Hong, Y. Wan, and A. K. Jain, "Fingerprint image enhancement: Algorithms and performance evaluation," *IEEE Transactions On PAMI*, vol. 20, no. 8, pp. 777–789, 1998.
- [35] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE Transaction on Image Processing*, vol. 9, no. 5, May 2000.
- [36] A. Lumini and L. Nanni, "Advanced methods for two class pattern recognition problem formulation for minutiae-based fingerprint verification." Article in Press, 2007.
- [37] E. Zhu, J. Yin, and G. Zhang, "Fingerprint matching based on global alignment of multiple reference minutiae," *Pattern Recognition*, vol. 38, pp. 1685–1694, 2005.
- [38] A. M. Bazen and R. N. J. Veldhuis, "Likelihood ratio-based biometric verification," *IEEE Transactions On Circuits And Systems For Video Technology*, vol. 14, no. 1, pp. 86–94, January, 2004.
- [39] K. A. Nagaty, "An adaptive hybrid energy-based fingerprint matching technique," *Image And Vision Computing*, vol. 23, pp. 491–500, 2005.
- [40] Y. He, J. Tian, L. Li, H. Chen, and X. Yang, "Fingerprint matching based on global comprehensive similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 6, June 2006.
- [41] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook Of Fingerprint Recognition*. Springer, 2003.
- [42] D. Wan and J. Zhou, "Fingerprint recognition using model-based density maps," *IEEE Transactions on Image Processing*, vol. 15, no. 6, June 2006.
- [43] L. O’Gorman and J. V. Nickerson, "An approach to fingerprint filter design," *Pattern Recognition*, vol. 22, no. 1, pp. 29–38, 1989.

BIBLIOGRAPHY

- [44] L. Nanni and A. Lumini, "A deformation-invariant image based fingerprint verification system," *Neurocomputing*, vol. 69, pp. 2336–2339, 2006.
- [45] T. Amornraksa and T. Tachaphetpiboon, "Fingerprint recognition using dct features," April 27, 2006.
- [46] J. V. Kulkarni, B. D. Patil, and R. S. Holambe, "Orientation feature for fingerprint matching," *Pattern Recognition*, vol. 39, pp. 1551–1554, 2006.
- [47] D. Maio and L. Nanni, "An efficient fingerprint verification system using integrated gabor filters and parzen window classifier," *Neurocomputing*, vol. 68, pp. 208–216, 2005.
- [48] F. Microsystems, "Interpreting fingerprint authentication performance," 2001. Technical White Paper.
- [49] "Neurotechnologija ltd. - software products." <http://www.neurotechnologija.com>.
- [50] A. Bovik, *The Image And Video Processing Handbook*. Academic Press, 1999.
- [51] K. Karu and A. Jain, "Fingerprint classification," *Pattern Recognition*, vol. 29, no. 3, pp. 389–404, 1996.
- [52] S. Acton, P. M. D., H. J.P., and C. B. A, "Oriented texture completion by am-fm reaction-diffusion," *IEEE Transactions On Image Processing*, vol. 10, no. 6, pp. 885–896, June, 2001.
- [53] A. R. Rao, *A taxonomy for texture description and identification*. Springer-Verlag Inc, 1990.
- [54] B.M.Mehtre, N.N.Murthy, S.Kapoor, and B.Chatterjee, "Segmentation of fingerprint images using the directional image," *Pattern Recognition*, vol. 20, no. 4, pp. 429–435, 1987.

BIBLIOGRAPHY

- [55] B.G.Sherlock, D.M.Monro, and K.Millard, "Fingerprint enhancement by directional fourier filtering," *IEE Proceedings On Vision, Image and Signal Processing*, vol. 141, no. 2, April, 1994.
- [56] T. Awcock, *Applied Image Processing*. The Macmillan Press Ltd, 1995.
- [57] "Digital image processing notes." <http://ct.radiology.uiowa.edu>.
- [58] R. Kimmel and A. Bruckstein, "Regularized laplacian zero crossings as optimal edge integrators," *International Journal Of Computer Vision*, vol. 53, no. 3, pp. 225 – 243, July-August, 2003.
- [59] V. Torre and T. A. Poggio, "On edge detection," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 8, pp. 147–163, 1986.
- [60] F. Destrempe and M. Mignotte, "A statistical model for contours in images," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 26, no. 5, May, 2004.
- [61] Y.-K. Lee and W. T. Rhodes, "Nonlinear image processing by a rotating kernel transformation," *Optics Letters*, vol. 15, no. 23, December, 1990.
- [62] K.V.Mardia, A. Baczkowski, X. Feng, and T. Hainsworth, "Statistical methods for automatic interpretation of digitally scanned fingerprints," *Pattern Recognition Letters*, vol. 18, pp. 1197–1203, 1997.
- [63] B.G.Sherlock and D.M.Monro, "A model for interpreting fingerprint topology," *Pattern Recognition*, vol. 26, no. 7, pp. 1047–1055, 1993.
- [64] P. R. Vizcaya and L. A. Gerhardt, "A nonlinear orientation model for global description of fingerprints," *Pattern Recognition*, vol. 29, no. 7, pp. 1221–1231, 1996.
- [65] J. Zhou and J. Gu, "A model based method for the computation of fingerprints' orientation field," *IEEE Transactions On Image Processing*, vol. 13, no. 16, June, 2004.

BIBLIOGRAPHY

- [66] E. Zhu, J. Yin, C. Hu, and G. Zhang, "A systematic method for fingerprint ridge orientation estimation and image segmentation," *Pattern Recognition*, vol. 39, pp. 1452–1472, 2006.
- [67] Y. Wang, J. Hu, and F. Han, "Enhanced gradient-based algorithm for the estimation of fingerprint orientation fields," *Applied Mathematics and Computation*, vol. 185, pp. 823–833, 2007.
- [68] L. Hong, A. Jain, S. Pankanti, and R. Bolle, "Fingerprint enhancement," *IEEE WACV*, May, 1996.
- [69] T. Kamei and M. Mizoguchi, "Image filter design for fingerprint enhancement," *International Symposium on Computer Vision*, 1995.
- [70] Z. Chao, L. Seah, V.M.Murukeshan, U. Dinish, and C.S.Tan, "An integrated phase-resolved fluorescence imaging system with sub-nanosecond lifetime resolution," *Optics and Laser Technology*, vol. 39, pp. 864–870, 2007.
- [71] L. Seah, U. Dinish, Z. Chao, and V.M.Murukeshan, "Phase-resolved fluorescence technique with heterodyne signal processing for imaging of latent fingerprints," *Journal of Modern Optics*, vol. 53, no. 12, pp. 1809–1817, August 15, 2006.
- [72] L. Seah, U. Dinish, S. Ong, Z. Chao, and V.M.Murukeshan, "Time-resolved imaging of latent fingerprints with nanosecond resolution," *Optics and Laser Technology*, vol. 36, pp. 371–376, 2004.
- [73] U. Dinish, *Investigation into phase-resolved optical technique for latent fingerprint detection and bioimaging*. PhD Thesis, School of Mechanical and Aerospace Engineering, Nanyang Technological University, 2005.

Publications

1. U.S. Dinish, Z.X.Chao, L.K.Seah, A.Singh and V.M.Murukeshan,"Formulation and implementation of a phase-resolved fluorescence technique for latent-fingerprint imaging: theoretical and experimental analysis,"*Applied Optics*, vol. 44, no. 3, pp. 297-304, 2005.
2. A.Singh, L.K.Seah and V.M.Murukeshan,"A novel orientation detection method based on quality measure," (*To be communicated*).
3. A.Singh, L.K.Seah and V.M.Murukeshan,"A scheme for detecting multiple orientations in an image," (*To be communicated*).

Appendix

Attached CD

- Partial Source Code For BiometricOCV Software in pdf format
- Matlab Scripts For Standard Algorithms

* Read the instructions.txt after opening the CD